Planned Perception within Concurrent Mapping and Localization

by

Michael P. Slavik

B.S. Systems Engineering
United States Naval Academy, 2000

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of
Master of Science in Electrical Engineering and Computer Science
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2002

Author .....
Department of Electrical Engineering and Computer Science
May 17, 2002

Certified by .....
William Kreamer
The Charles Stark Draper Laboratory, Inc.
Technical Supervisor

Certified by .....
John J. Leonard
Associate Professor, Department of Ocean Engineering
Thesis Advisor

Accepted by .....
Professor A. C. Smith
Chairman, Department Committee on Graduate Students

Planned Perception within Concurrent Mapping and Localization

by

Michael P. Slavik

Submitted to the Department of Electrical Engineering and Computer Science
on May 17, 2002, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

The fundamental requirement of truly autonomous mobile robots is navigation. Navigation is the science of determining one's position and orientation based on information provided by various sensors. Mobile robot navigation, especially autonomous vehicle navigation, is confronted with the problem of attempting to determine the structure of an *a priori* unknown environment, while at the same time using this information for navigation purposes. This problem is referred to as concurrent mapping and localization (CML). This thesis addresses the question of how to improve CML performance through smarter sensing strategies affecting robot behavior. Planned perception is the process of adaptively determining the sensing strategy of the mobile robot. The goal of integrating planned perception within concurrent mapping and localization is to attempt to answer the question of how a mobile robot should behave so as to attempt to optimize CML performance. This thesis demonstrates in simulation how the CML framework could be improved with planned perception by motivating changes in robot pose and hence, sensing locale.

Technical Supervisor: William Kreamer
Title: Member of the Technical Staff, The Charles Stark Draper Laboratory, Inc.

Thesis Advisor: John J. Leonard
Title: Professor of Ocean Engineering

Acknowledgments

I would like to thank my parents who have always loved, supported, and guided me throughout my life. Mom, I thank you for always being there and always believing in me. I wish everyone had a mother like you; you are the "bestest in the westest." Dad, no matter how many times I wanted to strangle you for saying, "Work hard," I thank you; you have taught me so much about life and leadership. Robbie and Joey, I can not express how much you both mean to me and how much you inspire me. I love you guys and look up to you (literally).

William Kreamer, I thank you for taking time out of your life to volunteer and take me on as your student. Your guidance at Draper Laboratory allowed me to finish this thesis. Without your help, I would still be searching for a thesis topic.

I would like to sincerely thank Professor John Leonard who took a chance and gave me the opportunity to work under him. For that, I will always be grateful. One of his biggest concerns was that I learned something. John, it is safe to say I learned more than I can express. You are a true teacher and leader.

My deepest and most sincere thanks goes to Dr. Paul Newman. Paul, you believed in me, inspired me, challenged me, and most of all, took the time to teach me and make sure I was always "with the woolidge." It was difficult in the beginning, but through your guidance and your support, I made it through. Without your help and confidence, this thesis would never have been written. I cannot thank you enough.

For the boys in the lab, it was fun. Jay Dryer, I appreciate all you the advice and guidance you have given me for the Navy and life in general. Sung Joon Kim, I thank you for your help and for introducing me to Diablo. Rick Rikoski, what can I say, you saved my thesis. The countless times you helped me will always be remembered.

Finally, my love and thanks go to Lauren. "My new wife," you are the light at the end of the tunnel; you are everything I have ever dreamed of. I cannot wait to spend the rest of my life with you.

"IT" is done. Thank God!

ACKNOWLEDGMENT

(17 May 2002)

This thesis was prepared at the Charles Stark Draper Laboratory, Inc., under Internal Company Sponsored Research and Development.

Publication of this thesis does not constitute approval by Draper or the sponsoring agency of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

Michael P. Slavik 17 *May* 2002

ASSIGNMENT

Draper Laboratory Report Number T-1429

In consideration for the research opportunity and permission to prepare my thesis by and at the Charles Stark Draper Laboratory, Inc., I hereby assign my copyright of the thesis to The Charles Stark Draper Laboratory, Inc., Cambridge, Massachusetts.

Michael P. Slavik 17 *May* 2002

Contents

[This page intentionally left blank]

List of Figures

List of Tables

[This page intentionally left blank]

Chapter 1

Introduction

Navigation is the science of determining one's position and orientation based on information provided by various sensors. Mobile robot navigation, especially autonomous vehicle navigation, is confronted with the problem of attempting to determine the structure of an *a priori* unknown environment, while at the same time using this information for navigation purposes. Overcoming this problem is essential for true autonomy [13, 26]. This problem is challenging because it must address two difficult issues simultaneously: navigation and mapping [2, 7, 27, 46]. Concurrent Mapping and Localization (CML) is the process of simultaneously building a map of the environment and using this map to obtain improved estimates of the location of the vehicle.

The fundamental requirement of truly autonomous mobile robots is navigation [21]; precision underwater vehicle navigation remains the principle obstacle to improved mobile robot and autonomous underwater vehicle (AUV) control [52]. Concurrent mapping and localization is intended to enable a mobile robot faced with the tasks of mapping and navigating to address these issues concurrently and reliably.

Autonomous vehicles must address the tasks associated with concurrent mapping and localization. This thesis addresses the question of how the incorporation of smarter sensing strategies will improve CML performance. Through the integration of planned perception with CML we will be motivated to change robot pose and hence, sensing locale. Adaptive sensing through planned perception will maximize

robot pose certainty through the re-observation of features of importance.

1.1 Navigation techniques used in CML

The localization problem is the specific problem of determining the location of a robot relative to a map [50]. Accurate localization forms the basis for most control and navigation decisions; without this feedback, control performance is limited [53]. Thus, reliable localization is an essential component of any autonomous vehicle system[24, 27, 51]. CML approaches the navigation problem with three primary representations: beacon-based navigation, relative navigation (dead reckoning and inertial navigation systems), and map-based navigation. Relative navigation is subject to external disturbances and uncorrectable drift allowing position errors to grow without bounds. However, bounded errors may be achieved through acoustic transponder resets. Regardless, CML has the potential to enable missions with bounded navigation errors without relying on *a priori* maps, acoustic beacons, or GPS resets [20, 27].

1.1.1 Dead reckoning and inertial navigation systems

Dead reckoning is the traditional and most common navigation technique. Estimates of vehicle position are obtained by integrating the vehicle velocity over time. Inertial navigation systems (INS) integrate the vehicle's acceleration twice to obtain new position estimates. Vehicle motion is sensed through gyroscopes and accelerometers. Starting with the most recent estimate of vehicle speed and direction, these integrated quantities are used to achieve updates of new position estimates.

Although dead reckoning is a common navigation technique, it has certain drawbacks. Navigation is limited by inaccuracies resulting from integration errors and system biases. Also, inertial navigation systems are expensive and consume much power. Relying exclusively on dead reckoning results in position error that grows without bound over time.

1.1.2 Beacon-based navigation

Beacons placed at known locations allow for vehicles to determine their position through triangulation [34]. The beacons emit pulses utilizing transducers or transducer arrays. Position fixes of vehicle location may be obtained by detecting these outgoing pulses and triangulating position location based on range and/or bearing measurements given *a priori* knowledge of beacon locations.

The satellite-based Global Positioning System (GPS) is the most prevalent beacon-based navigation system. This system, for many outdoor applications, provides a means for estimating position with a high accuracy. GPS signals are unable to be utilized indoors or underwater because of signal attenuation. Therefore, in the areas of indoor and underwater mobile robotics, GPS is not a reliable resource. Currently, two primary acoustic transponder systems are used in underwater navigation: long baseline (LBL) and ultra-short baseline (USBL) [34]. Both the USBL and LBL systems rely on accurate beacon positioning in order to obtain vehicle position relative to the transponders or beacons as shown in Figure1-1.

1.1.3 Map-based navigation

For some missions, beacons may be unavailable or impractical. However, if an *a priori* map of the environment is available, it may be possible to navigate relative to terrain. Map-based navigation is performed by correlating sensor data with an *a priori* map to deduce accurate localization. One such system is used by the U.S. Navy Tomahawk Cruise Missile (BGM-109). The Tomahawk's system couples inertial guidance with a terrain contour matching (TERCOM) guidance system [39]. By comparing a stored map to actual terrain measurements, it is possible to estimate the missile's position.

Unclassified, accurate, underwater maps are not often available. Even if a prior map exists, matching the sensed environment to that of the given prior map is a challenge. Map-based navigation has been applied by Carpenter and Medeiros [10]. They performed map-based navigation onboard an AUV with bathymetric data.

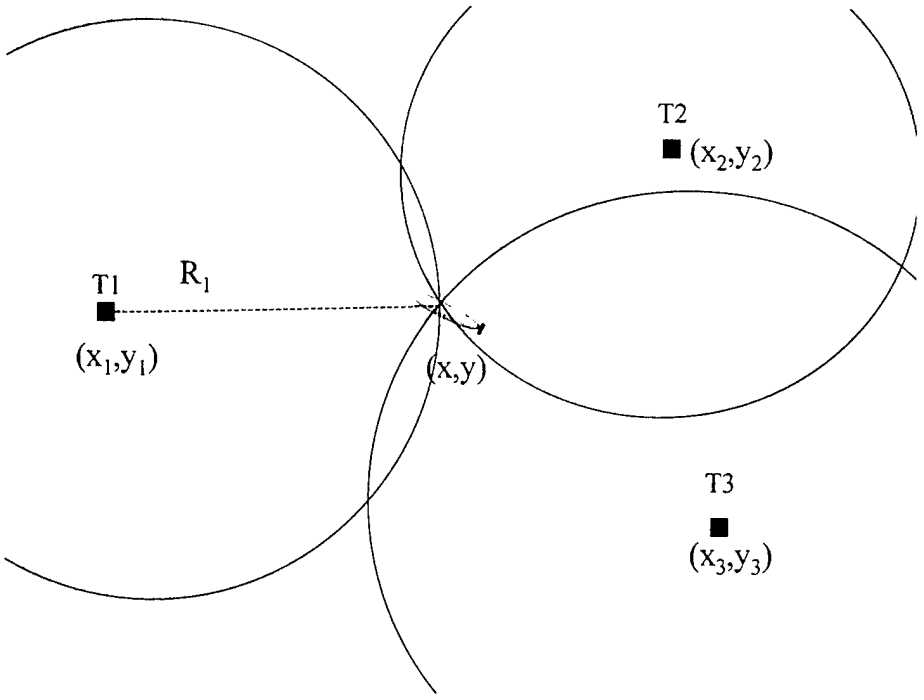


Figure 1-1: Basic LBL Description. The vehicle estimates its position by obtaining a "fix." A fix is obtained by triangulating range measurements received from each beacon.

1.2 Mapping techniques used in CML

Because *a priori* known maps of the environment are not always available, robots often must build such a map themselves. The task of building a map is estimating the locations of unknown objects in the environment [29]. Once a map of the environment is provided, accurate vehicle localization can be derived. Approaches to map creation can be categorized into three main areas: grid-based mapping, feature-based mapping, and topological mapping.

1.2.1 Grid-based mapping

Grid-based map representations, such as those implemented by Moravec [35] and Thrun *et al.* [48], represent the environment as an evenly-spaced grid. Individual cells that are certain to be occupied by a feature will be assigned a probability value of 1. Cells that are sure to be free of features are assigned probability values of 0. A

map of the environment is formed based on the assigned probabilities and is referred to as a certainty grid or occupancy grid. Mapping is performed by incorporating new measurements of the surrounding environment into the certainty grid. The probability values of the respective grid cells are adjusted according to the information and certainty of object locations obtained from the vehicle's measurements. The robot performs localization through map matching. Map matching is the process in which the robot creates a new local map and then compares this map to all previously constructed global maps. The best map match is found through the correlation of the local map with previous global maps. It is from this match that the new estimate of the robot's location is determined. Current state-of-the-art implementation of grid-based mapping has been performed by Thrun *et al.* [9].

The benefits of representing the environment using a grid-based map include simplicity of implementation and maintenance. However, this approach has a weak theoretical foundation and some feature-specific information (such as type of feature) is lost when features are assigned as probabilities to grid cells [21, 22, 29]. Other drawbacks to grid-based approaches are high storage requirements for data, difficulty in differentiating similar environments, and a high computational cost of localization [49].

1.2.2 Feature-based mapping

Rooted in the ideas of target tracking and surveillance [3], a geometric, feature-based representation of the map models the environment as a set of geometric primitives(such as points, lines, and planes) and builds a metrically accurate map to encode the landmark features [28, 29]. With the introduction of Stochastic Mapping (SM), Smith, Self, and Cheeseman [46] paved the way for future work in the field of feature-based mapping. Stochastic mapping, presented in more detail in Chapter 2, provides the theoretical foundation for the implementation of feature-based CML.

1.2.3 Topological mapping

Instead of representing the environment as a metrically accurate map, topological mapping generates graph-like descriptions of the environment. Kuipers and Byun [25] have used the topological approach by representing the environment as a graph of arcs and nodes [16]. Nodes and arcs represent different aspects of the environment map. Nodes represent easily distinguishable "significant places" in the environment. Arcs connect the nodes and represent the set of actions that connect the significant places. This topological model is behavior-based and utilizes reactive rules to move between the nodes, thus it is able to be used for route planning and obstacle avoidance. The drawbacks to using a topological approach to mapping are applications to larger environments and "significant place" recognition [25]. Because globally referenced, metrically accurate maps are a necessity for many operations, these drawbacks may pose a significant threat to achieving mission objectives [21].

1.3 Feature-based CML

Motivated by the need for accurate, reliable navigation systems for AUVs, this study of mobile robots pertains to autonomous vehicles overcoming the problem of operating in an unknown environment with imprecise navigation properties. The CML algorithm allows the vehicle, starting at an unknown location, to build a map of the previously unknown environment and then utilize that information to improve its own navigation estimate [44]. The research presented in this thesis concentrates on feature-based CML.

This thesis focuses on the feature-based approach of Stochastic Mapping (SM). Stochastic mapping [45] provides the theoretical foundation for the implementation of feature-based CML. First introduced by Smith, Self, and Cheeseman [46], stochastic mapping uses the extended Kalman filter (EKF) [4, 43] for state estimation. This algorithm stores estimates of robot and environmental feature locations and orientations in a single state vector along with an associated error covariance matrix representing the correlations between all mapped entities [41]. Features may be added to and re-

moved from the state vector and covariance matrix as time progresses; as new features are observed, the state space is augmented. The stochastic mapping algorithm is used to build and update a feature-based map of the environment from observations (sensor measurements) and from a model of vehicle dynamics/kinematics. Simultaneously, like the EKF, SM performs localization by updating vehicle position.

One challenge to stochastic mapping is data association. The data association problem consists of relating observations and measurements with the corresponding features [31, 44]. Data association and obtaining a correct solution is crucial because a misassignment will cause the filter estimates to diverge [37]. Solutions to the data association problem must be addressed to employ a stochastic mapping approach to CML [20].

Stochastic mapping and extensions to stochastic mapping have proven to yield valid solutions to the concurrent mapping and localization problem. Some solutions have expanded SM and incorporate track initiation, track deletion, and data association [17–19, 27, 44, 53]. Dissanayake [17] and Newman [38] provide an analysis of the performance of the stochastic mapping algorithm. These results show that a solution to the CML problem is possible; an autonomous vehicle located in an unknown position in an unknown environment can build a map, using only observations relative to its position, and simultaneously compute a bounded estimate of vehicle location.

The strengths and advantages of the feature-based approach of Stochastic Mapping to the CML problem are:

- Easily identifiable features in the environment are able to be extracted and mapped.
- A recursive solution to the navigation problem is provided.
- Consistent estimates for uncertainty in vehicle and feature locations are computed.
- SM can provide robust globally referenced navigation information.

- The SM approach to CML has the potential to enable autonomous robots to operate with bounded navigation errors without relying on acoustic beacons, *a priori* maps, or GPS updates.

1.4 Problem Statement

Stochastic mapping is a viable solution to the problem of operating in an unknown location in an unknown environment. The feature-based approach of CML utilizing stochastic mapping requires a mobile robot to take observations of its environment. The robot then maps its surrounding features and uses this map to navigate. Planned perception involves the focusing of sensory efforts to selected areas of interest. Areas of interest are defined as those areas and features which will improve the current navigation estimates and/or those which may contain new, useful features.

Planned perception is the process of adaptively determining the sensing strategy of the mobile robot. The goal of integrating planned perception with CML is to provide the mobile robot with a means to determine the optimal action given the current knowledge of robot pose (attitude and position), the environment, and sensors. Planned perception is a step in the direction of improving the overall CML framework. Strategic sensing will allow for a mobile robot to limit its area of sensing to those of interest or those that will provide a means of minimizing vehicle uncertainty.

By integrating CML with planned perception through smarter sensing strategies we will attempt to answer the question of how a mobile vehicle should behave so as to attempt to maximize CML performance. CML performance could be defined as discovering areas of interest, maximizing pose certainty, and maximizing map (feature) certainty.

This thesis will demonstrate in simulation how the CML framework could be improved through smarter sensing strategies and how a mobile vehicle should behave so as to attempt to optimize CML performance. The proposed approach will utilize a metric of feature quality and feature modality applied to robot sensing to motivate changes in robot behavior. We intend to show that CML performance, augmented

with smarter sensing strategies, will exhibit improved performance by motivating changes in robot orientation and sensing strategies.

1.5 Summary

The chapter presented the concept of concurrent mapping and localization as well as the idea of integrating planned perception with CML. The basic description of the current techniques of autonomous underwater vehicle mapping and navigation were presented and reviewed. This thesis focuses on the problem of extending feature-based SM and concurrent mapping and localization through the application of smarter sensing strategies which will affect the mobile vehicle's behavior.

1.6 Thesis contributions

This thesis makes the following contributions:

- A method for integrating planned perception within concurrent mapping and localization.
- An analysis of planned perception performance in simulation.

1.7 Thesis overview

This thesis presents an algorithm demonstrating how the CML framework could be improved through smarter sensing strategies and how a mobile vehicle should behave so as to attempt to optimize CML performance. The structure of this thesis is as follows:

Chapter 2 establishes the mathematical framework employed in the study of the concurrent mapping and localization problem.

Chapter 3 presents the theoretical contribution of this thesis as it addresses the integration of planned perception within the stochastic mapping algorithm.

Chapter 4 applies the integration of planned perception and CML. The design and results of the simulation are presented.

Chapter 5 summarizes the main contributions of this thesis and provides suggestions for future research.

Chapter 2

Stochastic Mapping

This chapter is concerned with the mathematical framework employed in this study of Concurrent Mapping and Localization. This framework was first introduced in a seminal paper [46] by Smith, Self, and Cheeseman and is known as Stochastic Mapping. Stochastic Mapping (SM) is simply a way of implementing feature-based concurrent mapping and localization utilizing an extended Kalman filter for state estimation [21].

This chapter reviews the stochastic mapping algorithm. It begins with a brief overview of the Kalman filter. Section 2.2 defines the state vector and covariance matrix which are used to describe the system behavior. Section 2.3 presents the models employed to represent a mobile robot and its environment in order to solve the CML problem. A generalized framework for the SM estimation process is then presented in Section 2.4. The chapter concludes with an analysis of structure of the CML problem and the convergence properties of the map and its steady-state behavior.

2.1 The Kalman Filter

In this section we will outline the Kalman filter and the extended Kalman filter. For a full derivation and a more detailed discussion refer to [4, 8, 23, 33, 43]. The Kalman filter is introduced because it serves as the premise for the extended Kalman filter

(EKF). The EKF is the observer [5] used in the stochastic mapping approach to CML.

2.1.1 The Kalman filter

The Kalman filter is a recursive least squares estimator. It is the optimal estimator if the dynamic model is linear and the noise processes are Gaussian. If the noise processes are non-Gaussian but the system is linear, the Kalman filter is the best linear estimator [4]. Without the loss of generality, in this discussion of the Kalman filter, a linear dynamic system and Gaussian noise processes are assumed.

The Kalman filter is a way of determining minimum mean-square error (MMSE) estimates using state-space methods. It is a procedure that uses the results of the previous time step to aid in obtaining the estimate at the current step of the process [8, 43]. The outline of the Kalman filter is seen in Figure 2-1.

At time step k the Kalman filter produces an MMSE estimate $\hat{\mathbf{x}}(k|k)$ of the state vector $\mathbf{x}(k)$. This estimate summarizes the information up to time step k - 1. Refinement of our estimate is obtained by fusing a prediction of the state estimate $\hat{\mathbf{x}}(k|k-1)$ with an observation $\mathbf{z}(k)$ of the state vector $\mathbf{x}(k)$. The recursion is completed when the estimated state $\hat{\mathbf{x}}(k|k)$ is predicted through a system model to produce a new estimate $\hat{\mathbf{x}}(k+1|k)$.

The Kalman filter is a recursive solution of the linear filtering problem. Despite the Kalman filter's efficiency, real navigation problems almost always involves a non-linear dynamic system. The extended Kalman filter is used in situations involving non-linear vehicle motion and measurement relationships.

2.1.2 The Extended Kalman filter

In any real navigation problem vehicle motion and the observation of features are almost always a non-linear processes. To compensate for the non-linearity of the dynamic system, two basic ways of linearizing the non-linearities are the *linearized Kalman filter* and the *extended Kalman filter* [8]. The linearized Kalman filter requires linearizing about some nominal trajectory in state space that does not depend

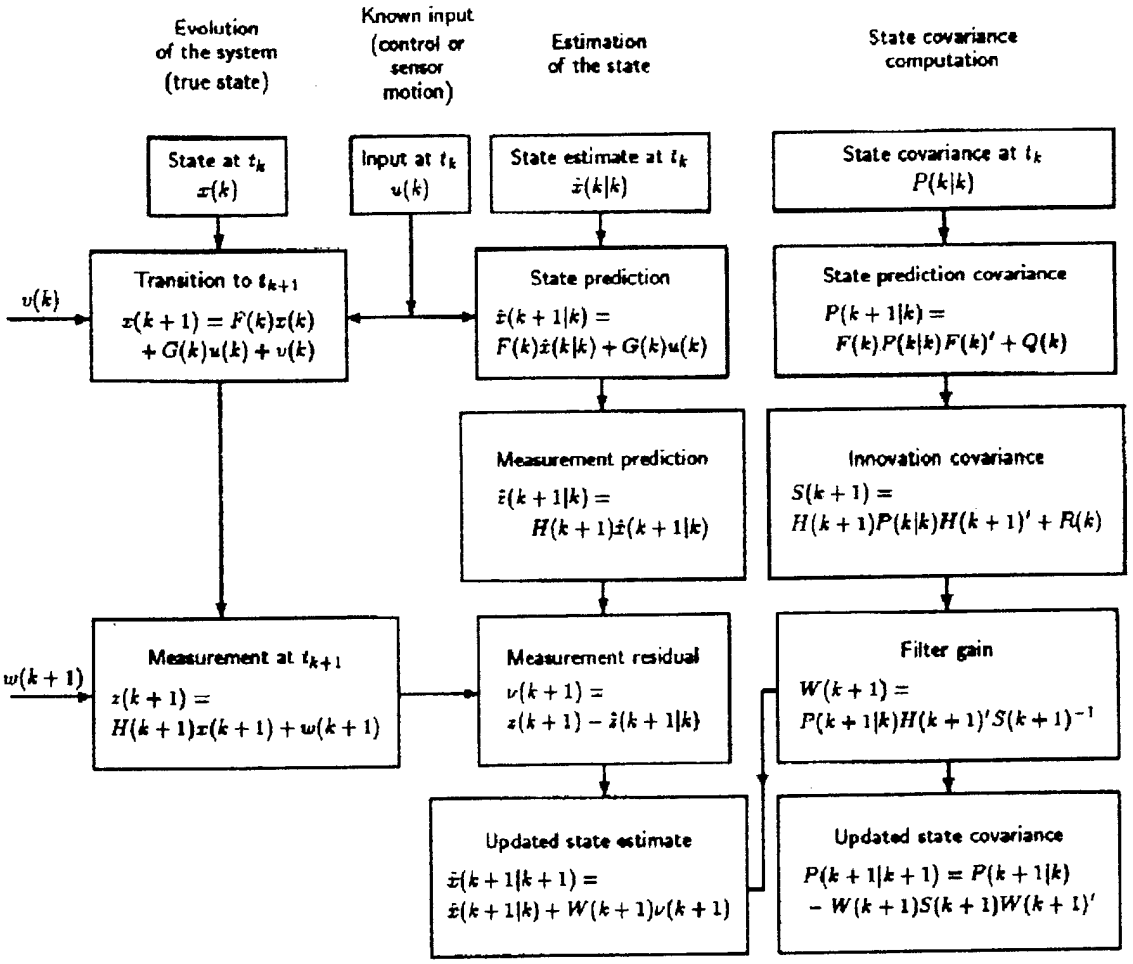


Figure 2-1: The Kalman Filter. *From Bar-Shalom and Li [4]*

on measurement data. The extended Kalman filter (EKF) linearizes about a trajectory that is updated continually with the state estimates resulting from observations. The EKF is the core method used in stochastic mapping. It is the technique of linearizing a non-linear dynamic system for use in a Kalman filter.

In order to circumvent the problem of operating with non-linearities, the non-linear models are approximated through a Taylor series expansion. The first order version of this expansion allows for the filter to be derived in the same manner as for the linear Kalman filter with the exception that the non-linear vehicle model and observation model are linearized [4].

This linearization is performed using Jacobians. The Jacobian of a function is a matrix of partial derivatives with respect to a vector. The Jacobian of a function $\mathbf{f}$ with vector $\mathbf{x}$ is defined as

$$\mathbf{f}_x = \nabla \mathbf{f} = \frac{\partial \mathbf{f}_m}{\partial \mathbf{x}_n} = \begin{bmatrix} \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_2} & \cdots & \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_n} \\ \frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_2} & \cdots & \frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{f}_m}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{f}_m}{\partial \mathbf{x}_2} & \cdots & \frac{\partial \mathbf{f}_m}{\partial \mathbf{x}_n} \end{bmatrix}. \tag{2.1}$$

Assuming that the approximation error of linearizing the non-linearities is small, the EKF can then be derived in the same manner as the linear case. The outline of the extended Kalman filter is seen in Figure 2-2.

It is the extended Kalman filter that forms the basis for the stochastic mapping algorithm. Smith, Self, and Cheeseman state that the reasons for using EKF are because of its simplicity in implementation, its similarity to the optimal linear filter (linear Kalman filter), and its ability to provide accurate estimates [46]. The EKF provides a means for which a mobile robot can perform relative measurements to features and obtain a bounded estimate of vehicle and landmark locations in a recursive fashion.

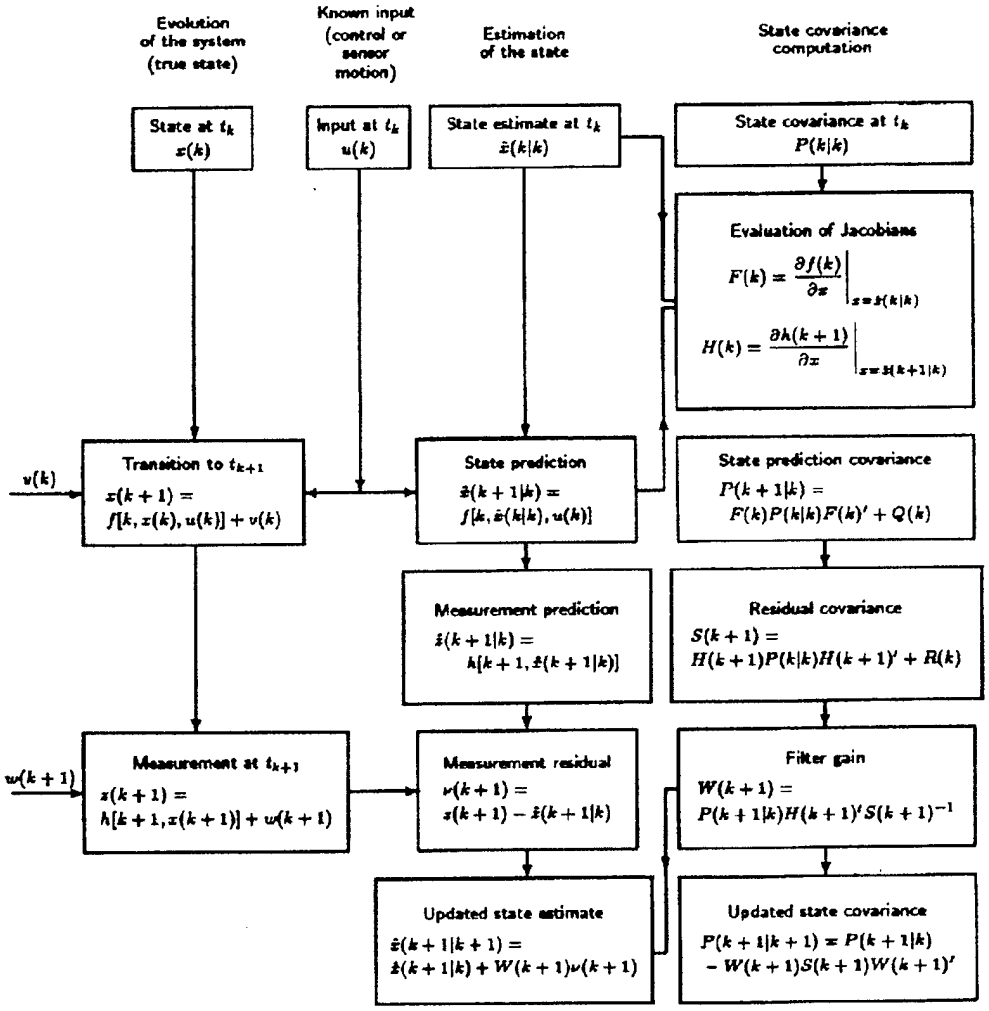


Figure 2-2: The Extended Kalman Filter. *From Bar-Shalom and Li [4]*

2.2 State vector and covariance matrix

Stochastic mapping represents the environmental map as a system state vector and an associated estimate error covariance. The state vector, $\mathbf{x}$, is defined as a combination of the vehicle states, $\mathbf{x}_v$, and feature states, $\mathbf{x}_f$. It stores the estimated state of the environment (feature locations) and the state of the vehicle (robot pose). The state of the system at time k can be represented by the augmented state vector, $\mathbf{x}(k)$. Where N_f describes the number of observed features at time k, $\mathbf{x}_{f_i}(k) = [x_{f_i} \; y_{f_i}]^T$, $i = 1, \ldots, N_f$, the system state vector is defined

$$\mathbf{x}(k) = \begin{bmatrix} \mathbf{x}_v(k) \\ \mathbf{x}_f(k) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_v(k) \\ \mathbf{x}_{f_1}(k) \\ \vdots \\ \mathbf{x}_{fN_f}(k) \end{bmatrix}. \tag{2.2}$$

The SM algorithm is a recursive estimation process that produces a MMSE estimate $\hat{\mathbf{x}}(k+1|k)$ of the state $\mathbf{x}$ given a sequence of observations up to time $k+1$, $\mathbf{Z}^k = \{\mathbf{z}(1), \ldots, \mathbf{z}(k)\}$. The filter fuses a prior estimate $\hat{\mathbf{x}}(k|k-1)$ with an observation $\mathbf{z}(k)$ of state $\mathbf{x}(k)$ at time k to produce an updated estimate $\hat{\mathbf{x}}(k|k)$. The state estimate is defined

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{E}[\mathbf{x}(k+1)|\mathbf{Z}^k] \tag{2.3}$$

$$= \begin{bmatrix} \hat{\mathbf{x}}_v(k+1|k) \\ \hat{\mathbf{x}}_{f1}(k+1|k) \\ \vdots \\ \hat{\mathbf{x}}_{fN_f}(k+1|k) \end{bmatrix}. \tag{2.4}$$

Throughout this thesis, the vehicle's state estimate will be defined by

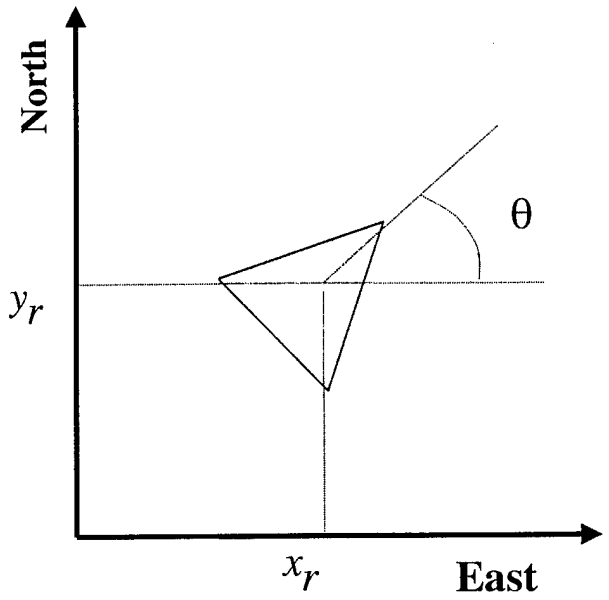


Figure 2-3: Definitions of the vehicle states used in the model

$$\hat{\mathbf{x}}_v = \begin{bmatrix} x_v \\ y_v \\ \theta \end{bmatrix} \tag{2.5}$$

to represent the vehicle's east position, north position, and heading as shown in Figure 2-3.

The estimated error covariance associated with the state vector is given by

$$\mathbf{P}(k|k) = \mathbf{E}[(\mathbf{x}(k) - \hat{\mathbf{x}}(k|k))(\mathbf{x}(k) - \hat{\mathbf{x}}(k|k))^T|\mathbf{Z}^k]. \tag{2.6}$$

This matrix defines the mean squared error and error correlations in each of the state estimates. The covariance matrix may also be written as

$$\mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}_{vv} & \mathbf{P}_{v1} & \cdots & \mathbf{P}_{vN_f} \\ \mathbf{P}_{1v} & \mathbf{P}_{11} & \cdots & \mathbf{P}_{1N_f} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{N_fv} & \mathbf{P}_{N_f1} & \cdots & \mathbf{P}_{N_fN_f} \end{bmatrix}_{k|k} . \tag{2.7}$$

The sub-matrices, $\mathbf{P}_{vv}$, $\mathbf{P}_{ii}$, and $\mathbf{P}_{vi}$ are the vehicle-to-vehicle, feature-to-feature, and

vehicle-to-feature covariances respectively. The vehicle ($\mathbf{P}_{vv}(k)$) and feature ($\mathbf{P}_{ii}(k)$) covariances are located on the main diagonal. They describe the uncertainty in the estimate of each state. The off-diagonals contain the vehicle-to-feature, ($\mathbf{P}_{vi}(k)$), and feature-to-feature, ($\mathbf{P}_{ij}(k)$), cross correlations. The off-diagonal terms describe the correlation between the uncertainties.

It is essential to maintain the cross correlations for two reasons. First, the information gained about one state can be used to improve the estimate of other correlated states. Second, maintaining cross correlation terms prevent the SM algorithm from becoming "overconfident." Being overconfident leads to incorrectly assuming features are independent when in fact, they are correlated [11].

Thus, the vehicle and map are represented by a single state vector $\mathbf{x}$ with estimate $\hat{\mathbf{x}}$ and an associated estimate error covariance $\mathbf{P}$. Given the definitions above, an EKF is employed to estimate the state and covariance given the measurements $\mathbf{z}$.

2.3 System models

The stochastic mapping algorithm depends on three key models: the vehicle model, the feature model, and the observation model. This section presents the general form of these models employed by this thesis. The vehicle model describes the kinematics and dynamics of the mobile robot. The feature model describes feature physics and relation to the environment. The observation model relates the observations to the state vector.

2.3.1 Vehicle model

The general form of a vehicle model (or process model) can be written as

$$\mathbf{x}_v(k+1) = \mathbf{f}_v[\mathbf{x}_v(k), \mathbf{u}_v(k+1), (k+1)] + \mathbf{v}_v(k+1). \tag{2.8}$$

This model attempts to capture the relationship between the vehicle's past state, $\mathbf{x}_v(k)$, and its current state, $\mathbf{x}_v(k+1)$, given a control input, $\mathbf{u}_v(k+1)$.

The vector $\mathbf{x}_v(k)$ is the vehicle state vector. It describes the state of the vehicle at time k. The vehicle state vector may contain any number of vehicle parameters. However, for the purpose of this thesis, the state vector is limited to defining the position and orientation of the vehicle in two dimensions.

The function $\mathbf{f}_v$ is the state transition matrix. It mathematically represents the mobile robot's dynamics, mobility, and kinematics. The discrete time vehicle model describes the transition of the vehicle state vector $\mathbf{x}_v$ from time k to time $k+1$.

The vector $\mathbf{u}_v(k+1)$ is the control input at time $k+1$. The control inputs in this thesis are vehicle speed and heading or steer angle, defining $\mathbf{u} = [v\ \theta]^T$.

The random vector $\mathbf{v}_v$ is the vehicle model process noise. It represents all of the noise and unmodelled aspects of vehicle behavior. This vehicle model process noise is assumed to be a stationary, temporally uncorrelated zero mean Gaussian white noise process. Because of this assumption, and the defined expectation in Equation 2.9, it can be shown that [8, 43]

$$\mathbf{E}[\mathbf{v}_v] = 0, \forall k \tag{2.9}$$

$$\mathbf{E}[\mathbf{v}_{vi} \cdot \mathbf{v}_{vj}^T] = \begin{cases} \mathbf{Q}_v(k) & \text{if } i = j = k \\ 0 & \text{otherwise} \end{cases}, \tag{2.10}$$

where $\mathbf{Q}_v(k)$ is the covariance of $\mathbf{v}_v$ at time k such that

$$\mathbf{Q}_v = \mathbf{Q} = \begin{bmatrix} v_v & 0 \\ 0 & \phi_v \end{bmatrix}, \tag{2.11}$$

and v_v and ϕ_v are the variances in error associated with velocity and heading, respectively. $\mathbf{Q}$ is the uncertainty in the process noise modelled with known variances.

The vehicle model used in this thesis is given by Equation 2.12

$$\mathbf{x}_v(k+1) = \mathbf{f}_v\mathbf{x}_v(k) + \mathbf{u}_v(k+1) + \mathbf{v}_v(k+1). \tag{2.12}$$

The dynamic model $\mathbf{f}_v$ is the defined as the state transition matrix for the vehicle

model. The vehicle model used in this thesis can be expanded and written as

$$\mathbf{x}_v(k+1) = \begin{bmatrix} x + v\Delta T\cos(\theta) \\ y + v\Delta T\sin(\theta) \\ \theta + \varphi \end{bmatrix}, \tag{2.13}$$

where ΔT is the change in time. This particular vehicle model is non-linear. The model will be linearized using its Jacobian evaluated at the vehicle state at time k such that

$$\nabla \mathbf{f}_v(k) = \begin{bmatrix} 1 & 0 & -v(k)\Delta T\sin\varphi(k) \\ 0 & 1 & v(k)\Delta T\cos\varphi(k) \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.14}$$

2.3.2 Feature model

A feature is a discrete, fixed, and identifiable landmark of the environment that can be consistently and reliably observed by the vehicle's sensors. Features can have many physical forms and can be both active features (artificial acoustic beacons) or passive features (points, planes, corners, poles).

Features are represented mathematically as a vector of parameters defining the landmark's properties. This thesis focuses on the simplest of all feature models, the stationary point landmark. The point feature is static and may be defined by two parameters specifying its position with respect to a global, 2-Dimensional coordinate frame. This type of feature is observable from any angle at any distance.

The i^{th} point feature in the environment is represented by the state estimate parameters defined as

$$\mathbf{x}_{f_i} = \begin{bmatrix} x_{f_i} \\ y_{f_i} \end{bmatrix}. \tag{2.15}$$

Since features are assumed to be stationary, there is no additive uncertainty term due to movement in the feature model. Thus, a trivial relationship exists between

the point feature state at time $k+1$ and k. Therefore, the point feature model can be represented by

$$\mathbf{x}_{f_i}(k+1) = \mathbf{x}_{f_i}(k) = \mathbf{x}_{f_i}. \tag{2.16}$$

2.3.3 Observation model

The general form for the observation model for the i^{th} feature is given by

$$\mathbf{z}_i(k) = \mathbf{h}_i[\mathbf{x}_v(k), \mathbf{x}_f(k), k] + \mathbf{w}_i(k). \tag{2.17}$$

The vector $\mathbf{z}_i(k)$ is the observation vector at time k. In this thesis, the observation vector consists of range and bearing measurements taken at time k. The range $\mathbf{z}_{r_i}(k)$ and bearing $\mathbf{z}_{\varphi_i}(k)$ measurements are of the i^{th} feature, with state $\mathbf{x}_{f_i}(k)$ relative to the robot's location $\mathbf{x}_v(k)$. The measurement vector is given by

$$\mathbf{z}_i(k) = \begin{bmatrix} \mathbf{z}_{r_i}(k) \\ \mathbf{z}_{\varphi_i}(k) \end{bmatrix}. \tag{2.18}$$

The function $\mathbf{h}_i$ is the observation model. The observation model relates the output of the sensor $\mathbf{z}_i$ to the state vector $\mathbf{x}$ when observing the i^{th} landmark.

The random vector $\mathbf{w}_i$ is the observation noise. All unmodelled sensor characteristics and noise corruption are represented in $\mathbf{w}_i(k)$. This observation error vector is again assumed to be a stationary, temporally uncorrelated zero mean random process. Because of this assumption and the defined expectation of Equation 2.19, [8, 43]

$$\mathbf{E}[\mathbf{w}_i] = 0, \forall k \tag{2.19}$$

$$\mathbf{E}[\mathbf{w}_{ii} \cdot \mathbf{w}_{ij}^T] = \begin{cases} \mathbf{R}_i(k) & \text{if } i = j = k \\ 0 & \text{otherwise} \end{cases} \tag{2.20}$$

where $\mathbf{R}_i(k)$ is the observation error covariance matrix of $\mathbf{w}_i$ at time k such that

$$\mathbf{R} = \begin{bmatrix} \mathbf{z}_{rw} & 0 \\ o & \mathbf{z}_{\varphi_w} \end{bmatrix}, \tag{2.21}$$

and $\mathbf{z}_{rw}$ and $\mathbf{z}_{\varphi_w}$ are the variances associated with the noise in range and bearing, respectively.

The observation model used in this thesis is given by Equation 2.22

$$\mathbf{z}_i(k+1) = \mathbf{h}_i[\mathbf{x}(k)] + \mathbf{w}_i(k) \tag{2.22}$$

$$= \mathbf{h}_i[\mathbf{x}_v(k), \mathbf{x}_{f_i}(k)] + \mathbf{w}_i(k), \tag{2.23}$$

where the defined observation matrix can be written and expanded as

$$\mathbf{h}_i[\mathbf{x}_v(k), \mathbf{x}_{f_i}(k)] = \begin{bmatrix} \sqrt{(\mathbf{x}_{f_i}(k) - x_v(k))^2 + (\mathbf{y}_{f_i}(k) - y_v(k))^2} \\ \arctan \frac{\mathbf{y}_{f_i}(k) - y_v(k)}{\mathbf{x}_{f_i}(k) - x_v(k)} - \theta_v \end{bmatrix}. \tag{2.24}$$

2.4 Stochastic mapping estimation process

Stochastic mapping (SM) is an EKF approach to the CML problem. This approach provides a recursive method for a mobile robot to yield bounded estimates of vehicle and landmark locations based on the information it obtained about the environment [29].

The CML problem addresses the idea of a vehicle operating in an unknown environment starting at an unknown location. The number of states needed to estimate the map cannot be fixed at the start of the mission because the number of features in the environment will not be previously known. Hence, the size of the state vector can not be pre-determined and must be changed during the estimation process.

Stochastic mapping considers CML as a variable-dimension state estimation problem. A single state vector is used to represent the map; the state vector contains estimates of vehicle location and environmental features. The state size increases or

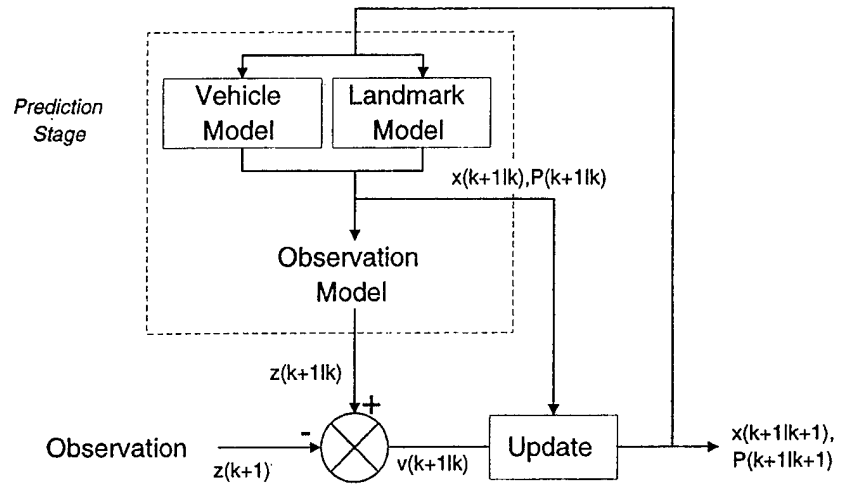


Figure 2-4: The role of the EKF in stochastic mapping.

decreases as features are added or removed from the map. An associated covariance matrix contains the uncertainties of the estimates and the correlations between the vehicle and feature estimates.

Stochastic mapping builds and updates a feature map of the environment from sensor measurements. Localization is performed simultaneously by updating the vehicle's position through an EKF. The approach is described in Figure 2-4.

Section 2.2 defined the state vector and associated covariance matrix used in representing the map in the SM approach to CML. The system models employed by SM were defined in Section 2.3. Sections 2.4.1 and 2.4.3 present the estimation process of SM. The data association problem is reviewed in Section 2.4.4 and Section 2.4.5 addresses how the representation of the map is augmented with the addition of new features.

2.4.1 Prediction

The prediction stage of the stochastic mapping algorithm uses the vehicle model described in Equation 2.12 and feature model described in Equation 2.16 to generate

a prediction of the state vector. This prediction is given according to

$$\begin{aligned}\hat{\mathbf{x}}(k+1|k) &= \mathbf{f}_v(k)\hat{\mathbf{x}}(k|k) + \mathbf{u}_v(k) \\ &= \begin{bmatrix}\hat{\mathbf{x}}_v(k+1|k) \\ \hat{\mathbf{x}}_f(k+1|k)\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{f}_v(k)\hat{\mathbf{x}}(k) + \mathbf{u}_v(k) \\ \hat{\mathbf{x}}_f(k)\end{bmatrix}.\end{aligned} \tag{2.25}$$

The state prediction Equation 2.26 is for a linear system, specifically a linear vehicle model. Our thesis utilizes a non-linear model as described in Section 2.3.1. Thus, using the linearized vehicle model, $\nabla\mathbf{f}_v$, in Equation 2.14, the state prediction is calculated

$$\hat{\mathbf{x}}(k+1|k) = \begin{bmatrix}\nabla\mathbf{f}_v(k)\hat{\mathbf{x}}(k) + \mathbf{u}_v(k) \\ \hat{\mathbf{x}}_f(k)\end{bmatrix}. \tag{2.26}$$

During the prediction stage, the feature state estimates remain unchanged because the features are assumed to be stationary.

Although features are assumed to be stationary, the vehicle is in motion and has an associated noise vector $\mathbf{v}_v$. This process noise is assumed to be uncorrelated and zero mean and can therefore be represented, based on Equations 2.9 and 2.10, with covariance $\mathbf{Q}$.

The prediction stage of the filter must also propagate the covariance matrix through the vehicle model. Defining $\nabla\mathbf{f}_v$ as the Jacobian of $\mathbf{f}_v$ evaluated at the estimated vehicle state $\hat{\mathbf{x}}_v$, the prediction of the covariance matrix is described by

$$\mathbf{P}(k+1|k) = \nabla\mathbf{f}_v(k)\mathbf{P}(k|k)\nabla\mathbf{f}_v{}^T(k) + \nabla\mathbf{f}_u(k)\mathbf{u}(k)\nabla\mathbf{f}_u{}^T(k) + \mathbf{Q}(k), \tag{2.27}$$

where $\nabla\mathbf{f}_v$ is the Jacobian of $\mathbf{f}_v$ evaluated at the estimated vehicle state $\hat{\mathbf{x}}_v$ defined in Equation 2.14, and $\nabla\mathbf{f}_u$ is the Jacobian of $\mathbf{f}_v$ evaluated with respect to the control input $\mathbf{u}$. The Jacobian $\nabla\mathbf{f}_u$ can be written and expanded as

$$\nabla \mathbf{f}_u = \begin{bmatrix} \Delta T \cos\theta & 0 \\ \Delta T \sin\theta & 0 \\ \Delta T \tan\varphi / L & \Delta T v \sec^2 \varphi / L \end{bmatrix}, \tag{2.28}$$

where L is the length of the vehicle. We will define the length of the vehicle to be 1 meter. The prediction stage of stochastic mapping may also be written

$$\begin{bmatrix} \mathbf{P}_{vv} & \mathbf{P}_{vf} \\ \mathbf{P}_{vf}^T & \mathbf{P}_{ff} \end{bmatrix}_{k+1|k} = \begin{bmatrix} \nabla \mathbf{f}_v(k) \mathbf{P}_{vv}(k|k) \nabla \mathbf{f}_v{}^T(k) + \nabla \mathbf{f}_u(k) \mathbf{u}(k) \nabla \mathbf{f}_u{}^T(k) + \mathbf{Q}(k) & \nabla \mathbf{f}_v(k) \mathbf{P}_{vf}(k|k) \\ (\nabla \mathbf{f}_v(k) \mathbf{P}_{vf}(k|k))^T & \mathbf{P}_{ff}(k|k) \end{bmatrix}. \tag{2.29}$$

The prediction of the estimated state and associated covariance matrix is made at each time step regardless of whether sensor measurements are taken. In the absence of a measurement, the prediction estimates the position of the mobile robot by propagating the state (essentially performing dead reckoning).

2.4.2 Observation

The fusion of measurements into state estimates begins by calculating a predicted observation at time k, termed $\hat{\mathbf{z}}_i(k)$. Applying the observation model Equation 2.22, a predicted measurement is calculated by

$$\hat{\mathbf{z}}(k+1|k) = \mathbf{h}[\hat{\mathbf{x}}_v(k+1|k), \hat{\mathbf{x}}_f(k+1|k)], \tag{2.30}$$

where $\mathbf{h}[\hat{\mathbf{x}}_v(k+1|k), \hat{\mathbf{x}}_f(k+1|k)]$ is defined by Equation 2.24.

Observations are received from sensors. These observation measurements must be associated with particular features in the environment. An association between the actual measurements with the predicted measurements is generated. The innovation, ν, is defined as the difference between the actual observations, $\mathbf{z}$, received from the systems sensors, and the predicted observations, $\hat{\mathbf{z}}$:

$$\nu(k) = \mathbf{z}(k) - \hat{\mathbf{z}}(k+1|k). \tag{2.31}$$

The innovation covariance, $\mathbf{S}(k)$ can then be calculated at time k. It is computed from the current estimate of the covariance matrix $\mathbf{P}(k+1|k)$, the covariance of observation noise $\mathbf{R}(k)$, and the Jacobian of the observation model $\mathbf{h}$ evaluated with respect to the estimated state vector, $\nabla\mathbf{h}_x$. The innovation covariance is then

$$\mathbf{S}(k+1) = \nabla\mathbf{h}_x(k)\mathbf{P}(k+1|k)\nabla\mathbf{h}_x{}^T(k) + \mathbf{R}(k+1). \tag{2.32}$$

The full measurement Jacobian $\nabla\mathbf{h}_x$ is a combination the Jacobians of the observation model evaluated with respect to the vehicle and feature states. This is noted because each observation is only a function of the feature being observed. Therefore, defining $\nabla\mathbf{h}_v$ and $\nabla\mathbf{h}_{fi}$ as the Jacobians of the observation model with respect to vehicle state and feature state i, respectively, the overall observation Jacobian can be shown in the form

$$\nabla\mathbf{h}_x(k) = [\nabla\mathbf{h}_v(k) \quad 0 \quad \ldots \quad 0 \quad \nabla\mathbf{h}_{fi}(k) \quad 0 \quad \ldots \quad 0]. \tag{2.33}$$

For the purposes of this thesis, the Jacobian $\nabla\mathbf{h}_v$ is

$$\nabla\mathbf{h}_v = \begin{bmatrix} \frac{x_v - \mathbf{x}_{f_i}}{r} & \frac{y_v - \mathbf{y}_{f_i}}{r} & 0 \\ \frac{\mathbf{y}_{f_i} - y_v}{r^2} & -\frac{\mathbf{x}_{f_i} - x_v}{r^2} & -1 \end{bmatrix}, \tag{2.34}$$

and the Jacobian $\nabla\mathbf{h}_{fi}$ is

$$\nabla\mathbf{h}_{fi} = \begin{bmatrix} \frac{\mathbf{x}_{f_i} - x_v}{r} & \frac{\mathbf{y}_{f_i} - y_v}{r} \\ -\frac{\mathbf{y}_{f_i} - y_v}{r^2} & \frac{\mathbf{x}_{f_i} - x_v}{r^2} \end{bmatrix}, \tag{2.35}$$

where r is the range defined, $r = \sqrt{(\mathbf{x}_{f_i}(k) - x_v(k))^2 + (\mathbf{y}_{f_i}(k) - y_v(k))^2}$.

2.4.3 Update

The observation $\mathbf{z}_i(k+1)$ is used to update the state and covariance predictions yielding new estimates at time $k+1$. The new state update $\hat{\mathbf{x}}(k+1|k+1)$ and updated state covariance $\mathbf{P}(k+1|k+1)$ are

$$\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{W}(k+1)\nu(k+1) \quad (2.36)$$

$$\mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{W}(k+1)\mathbf{S}(k+1)\mathbf{W}^T(k+1), \quad (2.37)$$

where the gain matrix is given by

$$\mathbf{W}(k+1) = \mathbf{P}(k+1|k)\nabla\mathbf{h}_x{}^T(k+1)\mathbf{S}^{-1}(k+1). \quad (2.38)$$

The pose estimate and associated errors are updated via the weighting factor of the gain matrix. The weighting factor is proportional to $\mathbf{P}(k+1|k)$ and inversely proportional to the innovation covariance $\mathbf{S}(k+1)$. Thus, if the innovation covariance is large compared to the state covariance, the weighting factor approaches zero and the measurement has little affect on updating the state estimate. Conversely, if the prior state covariance is large compared to the innovation covariance, the weighting factor approaches identity and the state is updated taking into account nearly all of the difference between the measurement and expected value [43, 46].

2.4.4 Data association

The data association problem consists of relating observations and measurements to corresponding features included in the map. Data association algorithms are motivated by the desire to assign measurements to the features from which they originate. The most common method for performing data association are nearest neighbor techniques [21, 27, 30, 38]. This thesis applies the nearest neighbor gating data association algorithm.

This section describes the technique which allows an observation, $\mathbf{z}$, to be associated with a landmark, $\mathbf{x}_{f_i}$. It relies on the innovation, ν, and innovation covariance, $\mathbf{S}$. Given an observation $\mathbf{z}(k)$, according to Equations 2.31 and 2.32, the Mahalanobis distance is defined as γ where

$$\gamma = \nu^T(k)\mathbf{S}^{-1}(k)\nu(k) \tag{2.39}$$

and is tested

$$\gamma \leq \gamma_{gate}. \tag{2.40}$$

The quantity γ has a χ^2 probability distribution and can be used to accept or reject a particular association given a confidence level, or "validation gate," γ_{gate}. For a system with 2 degrees of freedom, a value of 9.0 yields the region of minimum volume that contains the observation with a probability of 98.9% [3]. It is this γ_{gate} that the Mahalanobis distance is gated against

$$\gamma_{gate} = 9.0 \tag{2.41}$$

$$\gamma \leq 9.0 \tag{2.42}$$

The test is performed for all known features, $i = 1 \ldots N_f$, (all landmarks previously mapped). For all measurements $\mathbf{z}$ that can be potentially associated with feature i, γ_i is calculated and tested according to Equation 2.40. The validation of this equation defines where a measurement is expected to be found. If some $\gamma_i \leq \gamma_{gate}$, the observation is used to update the state estimates. However, if an observation does not gate with any existing feature, it can be used to initialize a new feature into the map. The next section describes how a feature is initialized and augmented into the SM state space representation.

2.4.5 Feature initialization

The stochastic mapping algorithm allows for features to be added to and removed from the state vector and covariance matrix as time progresses. When a new feature is observed by measurement $\mathbf{z}$, it must be initialized and added to the map; as new features are observed the state space is augmented. The initial estimate of the new

feature state is

$$\hat{\mathbf{x}}_{f_{N_f+1}}(k) = \mathbf{g}[\hat{\mathbf{x}}_v(k), \mathbf{z}(k)], \tag{2.43}$$

where N_f represents the number of known features up to the current time step. The feature initialization model $\mathbf{g}$ maps the current vehicle estimate and observation to a new feature estimate. These new state estimates are appended to the estimated state vector as a new feature estimate consistent with the feature model, Equation 2.16. The estimated state vector is augmented to become

$$\hat{\mathbf{x}}(k+1|k+1) = \begin{bmatrix} \hat{\mathbf{x}}(k+1|k) \\ \hline \hat{\mathbf{x}}_{f_{N_f+1}} \end{bmatrix}. \tag{2.44}$$

The uncertainty of the new feature estimates must also be initialized in the covariance matrix based on the new observation. This observation was taken relative to the robot. Clearly, the uncertainty in the new feature's estimated location is correlated with the uncertainty in the robots's position. This uncertainty is therefore not only correlated with the vehicle, but also correlated with the other map state estimates (features).

In order to account for the correlation of the estimated new feature and the previously mapped state estimates, the covariance matrix must be augmented. The augmented covariance matrix is

$$\mathbf{P}(k+1|k+1) = \left[\begin{array}{c|c} \mathbf{P}(k+1|k) & \mathbf{B}^T \\ \hline \mathbf{B} & \mathbf{A} \end{array}\right], \tag{2.45}$$

where $\mathbf{P}(k+1|k)$ is the estimate in the covariance matrix obtained from the prediction and $\mathbf{B}$ and $\mathbf{A}$ are defined as

$$\mathbf{B} = \nabla \mathbf{g}_x \mathbf{P}(k+1|k), \tag{2.46}$$

$$\mathbf{A} = \nabla \mathbf{g}_x \mathbf{P}(k+1|k) \nabla \mathbf{g}_x{}^T + \nabla \mathbf{g}_z \mathbf{R} \nabla \mathbf{g}_z{}^T, \tag{2.47}$$

and $\mathbf{R}$ is the covariance error in observations defined in Equation 2.20. The symbols $\nabla \mathbf{g}_x$ and $\nabla \mathbf{g}_z$ are the Jacobians of the feature initialization model $\mathbf{g}$ with respect to the state and observation, respectively. The feature initialization model $\mathbf{g}$ can be written as

$$\begin{bmatrix} \mathbf{x}_{f_{new}} \\ \mathbf{y}_{f_{new}} \end{bmatrix} = \begin{bmatrix} x_v + \mathbf{z}_r \cos(\theta + \mathbf{z}_\varphi) \\ y_v + \mathbf{z}_r \sin(\theta + \mathbf{z}_\varphi) \end{bmatrix}. \tag{2.48}$$

Thus $\nabla \mathbf{g}_x$ and $\nabla \mathbf{g}_z$ can be computed as

$$\nabla \mathbf{g}_x = \begin{bmatrix} 1 & 0 & -\mathbf{z}_r \sin(\theta + \mathbf{z}_\varphi) \\ 0 & 1 & \mathbf{z}_r \cos(\theta + \mathbf{z}_\varphi) \end{bmatrix}, \tag{2.49}$$

$$\nabla \mathbf{g}_z = \begin{bmatrix} \cos(\theta + \mathbf{z}_\varphi) & -\mathbf{z}_r \sin(\theta + \mathbf{z}_\varphi) \\ \sin(\theta + \mathbf{z}_\varphi) & \mathbf{z}_r \cos(\theta + \mathbf{z}_\varphi) \end{bmatrix}. \tag{2.50}$$

This framework allows for a variable-dimension state representation of the vehicle and environment while still providing a recursive solution to the navigation problem.

2.5 Structure of the CML problem

This section discusses conclusions of work performed by Newman [38] and presented in Dissanayake *et al.* [17] that describe the convergence properties of the environmental map and its steady state behavior. These results provide crucial insight into the CML problem. The three convergence theorems are:

First Convergence Theorem: The determinant of any sub-matrix of the map covariance matrix decreases monotonically as successive observations are made.

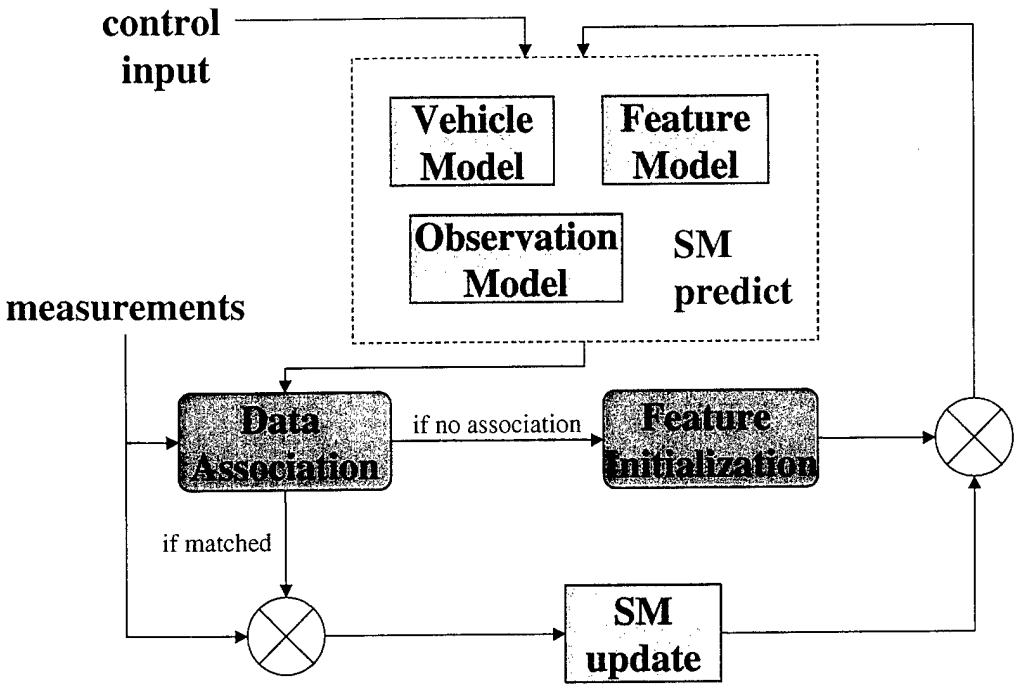


Figure 2-5: The Stochastic Mapping Algorithm

Second Convergence Theorem: In the limit the landmark estimates become fully correlated.

Third Convergence Theorem: In the limit, the lower bound on the covariance matrix associated with any single landmark estimate is determined only by the initial covariance in the vehicle estimate $\mathbf{P}_{0v}$ at the time of the first sighting of the first landmark.

The first theorem provides the following information. The uncertainty of a state estimate can be measured quantitatively by taking the determinant of a state covariance sub-matrix. The determinant of any sub-matrix of the map is a measure of the volume of uncertainty associated with the respective state estimate. This theorem states that the total uncertainty of the state estimate does not increase during the update step of the stochastic mapping algorithm; error is added during the prediction step and subtracted during the update step (refer to Sections 2.4.1 and 2.4.3). Thus, the error estimates of absolute location of features diminishes as successive measurements are made.

The second theorem proves that error estimates for the vehicle and feature states

decrease as more and more observations are made. As the number of observations approaches infinity, features become more correlated. Thus, given the exact location of one landmark, the location of all other landmarks can be deduced with absolute certainty. Therefore, the map becomes fully correlated.

The third theorem states that the absolute error for a single feature or vehicle can never be lower than the absolute vehicle error that was present at the time the feature was first initialized into the map. This proves that the error for a feature not be less than the error of the vehicle that was present at the time when the feature was first observed.

2.6 Summary

This chapter presented the stochastic mapping algorithm which is described in Figure 2-5. This algorithm serves as the mathematical framework for this feature-based approach to CML. It is the estimation algorithm that will be employed in this thesis. A brief overview of the work-horse of the SM algorithm, the EKF, was presented. The steps involved in SM were then presented as well as methods in which to account for data association and initializing features into the map representation.

Chapter 3

Planned Perception

This chapter contains the theoretical contribution of this thesis. It presents a technique for adaptive concurrent mapping and localization (CML) based on motivating changes in robot behavior in an attempt to improve CML performance. This technique will be evaluated by integrating planned perception with CML.

3.1 Introduction

Described in Chapter 2, SM is a viable solution to the problem of operating in an unknown location in an unknown environment. CML requires a mobile robot to take observations of objects in its environment. It maps the surrounding features based on these measurements and then uses this map for navigation.

Planned perception is the process of adaptively determining the sensing strategy of the mobile robot. The goal of integrating smarter sensing strategies within CML is to provide the mobile robot with a means to determine the optimal action given the current estimates of the vehicle and map. Planned perception involves the focusing of sensory efforts to selected areas of importance. Areas of importance are defined as those areas and features which will improve the current navigation estimates and/or those which may contain new, useful features. By integrating CML with planned perception we will attempt to answer the question of how a mobile robot should behave so as to attempt to maximize CML performance.

Accurate information regarding the pose of the vehicle is vital for mission success. This drives the basis for our implementation of adaptive CML; planned perception is formulated so as to maximize vehicle certainty. The SM algorithm provides an inherent coupling (through cross-correlations) of the estimated states and associated uncertainties. Maintain of these cross-correlations is important to the CML process [17, 38] as seen by the theorems described in Section 2.5.

The CML process is a map-building system. What is important in a map-building system is maintaining the consistency and integrity of the whole map and the estimate of the mobile robot's position within it [15].

Since we aim to minimize vehicle uncertainty, our proposed approach chooses a landmark, from the available set of features, whose re-observation will best improve the robot's estimated state. This choice is based on criteria that will be described in Section 3.4.

Section 3.2 presents scenarios that demonstrate why planned perception in CML is of interest. Section 3.3 reviews previous research in the area of adaptive sensing strategies. The proposed algorithm for integrating planned perception and CML is presented in Section 3.4. A summary of the chapter is provided in Section 3.5.

3.2 Motivation

Why is it important to adaptively choose a sensing and motion strategy for a mobile vehicle? Motivation behind integrating planned perception and CML can be seen through the following examples. These examples give an insight into the possible applications and motivating scenarios behind planned perception.

Efficiency

Planned perception and adaptive sensing techniques can be used to limit sensing to selected regions of interest, benefitting navigation precision and dramatically reducing computational requirements [21]. Planned perception within the CML algorithm will benefit mobile robots because it will reduce location estimation errors and reduce the

time and energy required to achieve a desired location accuracy. Thus, we will be improving the efficiency of the CML process for mobile robots [20].

Echo-location

The behavior of dolphins and bats show how adaptive sensing can be beneficial. These animals use sonar systems from which they can establish and maintain "contact" by receiving echoes from features within the environment [20]. Bats and dolphins are constantly adaptive their movements during their "flight paths." This allows them to get a better understanding of the environment by obtaining sonar responses from multiple vantage points [1].

This process is similar to the way a human perceives the world. One's eyes are constantly moving, constantly adjusting, and constantly focusing on different objects to obtain a "picture" of the world. Thus, observing the world through planned perception is inherent in both humans and bats and dolphins; humans use vision and dolphins and bats use sonar.

Military applications

There is great potential for military application of autonomous underwater vehicles (AUVs). Examples of importance to the military, especially the U. S. Navy, are those associated with specific aspects of Naval Doctrine: Command, Control, and Surveillance, Power Projection, Force Sustainment, and Battlespace Dominance [36]. Command, Control, and Surveillance is defined as the gathering, processing, and distribution of information that is vital to military operations. Power Projection is the ability to respond to any call-to-duty around the world and to take the fight to the enemy. Force Sustainment is defined as the capability to sustain our forces at home and abroad through effective operation and supply. Battlespace Dominance is maintaining superiority of the always shifting, fluid, zones of military areas of operation. AUVs can provide the Navy with a means of enhancing the above missions through their inherent stealth and by reducing the risk of losing human life. AUVs can help fulfill the needs of the Navy because by shaping the battlespace in areas

denied to traditional maritime forces.

Unlike current manned platforms, AUVs possess the capabilities of providing autonomy and enhanced deployability while keeping men out of harm's way. These skills are desirable in any maritime operation. AUVs may be launched from multiple platforms such as shore facilities, ships, submarines, and aircrafts. Autonomy allows for independent operation away from manned forces. This, in turn, allows for manned platforms to perform more complex tasks while extending the "reach" of their forces.

The ability of an AUV to perform planned perception would allow the vehicle to strengthen its ability to navigate reliably while accomplishing its mission. Planned perception allows for a vehicle to focus its sensory efforts, focusing on features of good quality, in order to maintain an accurate estimate of vehicle state. Thus, planned perception may enable a vehicle to better perform missions such as antisubmarine warfare (ASW), intelligence, surveillance, and reconnaissance (ISR), and undersea search and survey.

ISR missions may be enhanced by planned perception aboard AUVs. Due to their smaller size, AUVs provide superior maneuverability and are able to operate in shallow waters while still providing the ability to avoid detection. Thus, AUVs may be used for shallow water reconnaissance. Planned perception integrated with CML would allow a vehicle to focus on features of good quality. This would be beneficial in MCM (mine counter measure) missions because it provides a means of clandestine detection and mapping of a mine field.

The integration of CML with an adaptive sensing strategy would also enhance ASW missions. Planned perception would allow an AUV to focus its sensory efforts on a submarine while providing a means to determine how to maneuver to acquire the maximum information about this target. AUVs could also serve as an autonomous weapons platform. Smarter sensing strategies may enable the vehicle to focus on its target and increase the probability of achieving mission success.

One of the greatest fears of any submariner is the fate suffered by those lost in submarines such as the USS Scorpion, the USS Thresher, and the Russian Submarine Kursk [12, 40]. Currently, the ability to detect submarines on the ocean floor (a subset

of undersea search and survey) is provided through either manned vehicles or the use of side-scan sonar deployed on towed arrays or on remotely operated vehicles (ROVs). However, AUVs utilizing planned perception may provide a greater advantage to locating objects, such as submarines and shipwrecks, on the bottom of the ocean. Integrating planned perception with CML would enable AUVs to detect these features and classify them as features of importance. Therefore, in this theater of operation, an AUV would be able to maneuver itself through adaptive sensing strategies to maximize the information gained from the environment.

Closing the loop

A crucial measure of success is the frequency with which the vehicle recognizes its initial location after it *closes the loop*. Closing-the-loop defines the problem for CML as it is the most important measure of how well the vehicle understands the environment. It is defined as the ability to recognize that the vehicle is back in the same place and that it is re-observing a feature it has previously sensed and mapped. Planned perception would enable the vehicle to increase its chances of closing the loop by relying on its ability to maneuver through the area of operation using adaptive sensing.

3.3 Previous Work

Strategic sensing allows for a mobile robot to limit its area of sensing to those of interest. This section reviews work related to the focus of this thesis: the demonstration of how the CML framework could be improved through smarter sensing strategies and how a mobile vehicle should behave so as to attempt to optimize CML performance.

Planned perception has become a popular research topic. The idea of planned perception has been investigated in the area of robotics. Research synonymous with planned perception include active perception [47], active vision [14, 15], directed sensing [29], adaptive sensing [20, 21], adaptive sampling [6], and sensor management [32]. A common theme among the various adaptive sensing strategies has been choosing

the action which maximizes the information gained by the robot after evaluating the different sensing actions the robot can perform. While implementing this concept it is imperative to develop a quantitative method of evaluating the various future sensing actions of the robot [21].

Bellingham and Willcox [6] and Singh [42] have applied the concept of strategic sensing to marine robotics. Singh formulated and implemented adaptive sensing on the Autonomous Benthic Explorer. Bellingham and Willcox investigated dynamic oceanographic phenomena for AUVs. Also, Manyika and Durrant-Whyte [32] developed a method for implementing an information theoretic approach to planned perception. This method was applied to a mobile robot operating indoors, given an *a priori* map, and sensing using multiple scanning sonars.

Our approach is motivated by Feder's work [21] and is closest to Davison's implementation of active vision [14, 15]. Feder, Leonard, and Smith [20] introduced a metric for adaptive sensing defined in terms of Fisher information. The goal of this approach is to determine the action that maximizes the total knowledge about the system in the presence of uncertainties in navigation and observations. The next action of the robot is chosen to attempt to maximize the robot's information about its location and all the features' locations (the map) [20]. The action that maximizes the information is expressed as

$$\mathbf{u}_k = arg_u max\ \mathbf{I}_{k+1|k+1} = arg_u min\ \mathbf{P}_{k+1|k+1} \tag{3.1}$$

where $\mathbf{u}$ is the control input, $\mathbf{I}$ is the Fisher information, and $\mathbf{P}$ is the error covariance [21]. A cost function is defined as

$$\mathbf{C}(\mathbf{P}) = \pi\sqrt{det(\mathbf{P}_{vv})} + \pi\sum_{i=1}^{N}\sqrt{det(\mathbf{P}_{ff})} \tag{3.2}$$

The action to take is given by evaluating Equation 3.1 using the metric in Equation 3.2 [21]. The approach was demonstrated via simulation, underwater sonar experiments, and in-air sonar experiments [20, 21]

Davison implemented active vision on a mobile robot [14, 15]. His work provides a

way for a mobile robot, with active camera sensors, to track and fixate features over a wide field of view. Stationary point features are used to observe and obtain estimates of robot state utilizing a Kalman filter approach [15]. Davison and Murray propose three questions, or issues, that need to be considered to maintain the map [14]:

1. Which of the current set of known features should be tracked?

2. Is it time to label any features as not useful and abandon them?

3. Is it time to look for new features?

The third question is addressed when less than two features are currently visible. Question 2 is answered based on a method that abandons features if more than half of at least 10 attempts to observe that feature fail when it should be visible.

To answer the first question, Davison presents a way [14, 15] to calculate the volume of error (termed $\mathbf{V_s}$) in 3-Dimensional space, for each feature. The quantity $\mathbf{V_s}$ is based on a metric using eigenvalues and is evaluated for each feature currently visible. The error with the highest value is selected to be measured. The goal is to observe the feature with the greatest uncertainty in order to "squash" the total uncertainty [14].

3.4 Planned Perception

This section reviews the concept of planned perception and derives a method for integrating planned perception and concurrent mapping and localization within the stochastic mapping algorithm.

3.4.1 Concept

SM provides a means to perform CML. The map's information, consisting of the robot and features, is maintained as described in Chapter 2. The estimated state vector, $\mathbf{x}$, and associated covariance matrix, $\mathbf{P}$, are defined

$$\hat{\mathbf{x}}(k+1|k) = \begin{bmatrix} \hat{\mathbf{x}}_v(k+1|k) \\ \hat{\mathbf{x}}_{f1}(k+1|k) \\ \vdots \\ \hat{\mathbf{x}}_{fN_f}(k+1|k) \end{bmatrix} \tag{3.3}$$

$$\mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}_{vv} & \mathbf{P}_{v1} & \cdots & \mathbf{P}_{vN_f} \\ \mathbf{P}_{1v} & \mathbf{P}_{11} & \cdots & \mathbf{P}_{1N_f} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{N_fv} & \mathbf{P}_{N_f1} & \cdots & \mathbf{P}_{N_fN_f} \end{bmatrix}_{k|k} . \tag{3.4}$$

The sub-matrix $\mathbf{P}_{vv}$ represents the uncertainty in the vehicle estimates. Our goal is to minimize vehicle uncertainty. The approach relies on the convergence properties of CML reviewed in Section 2.5 and proved in [17, 38].

The determinant of the state covariance matrix, Equation 3.4, is a measure of the volume of the uncertainty ellipsoid associated with the state estimate [17]. Thus, the determinant of the sub-matrix $\mathbf{P}_{vv}$ represents the uncertainty ellipsoid associated with the vehicle state estimate $\hat{\mathbf{x}}_v$. It is this determinant that we will attempt to minimize through adaptively determining the sensing strategy of the mobile robot.

A mobile robot is given the task of performing a specified mission. Performing this mission is defined as *exploration*. Exploration is the process of attempting to perform the mission objective. Mission objectives vary from scenario to scenario but some examples are: navigating between waypoints [21], traversing the corridors of a building [15], and detecting objects on the seabed [27]. These are only a few examples of mobile robot mission objectives, the possible missions a robot may be asked to perform are innumerable.

Despite the specific mission, one aspect of exploration is constant; the mobile robot must accurately navigate to accomplish the objective. Reliable navigation requires good estimates of the vehicle state. Our proposed approach to incorporating planned perception in the SM algorithm focuses on maximizing vehicle certainty, therefore benefitting navigation precision.

While a vehicle is "confident" in its estimates, the determinant of the sub-matrix $\mathbf{P}_{vv}$ is relatively small. Therefore, the mission can be carried out as planned. However, when the uncertainty in vehicle estimates grows, the determinant of $\mathbf{P}_{vv}$ also grows. Thus, the vehicle becomes less certain of its overall state. It is at this instant, when the determinant of $\mathbf{P}_{vv}$ exceeds some threshold, that our method of planned perception motivates changes in robot pose and hence, sensing locale. The method of switching operating mode from *exploration* to *localization* (planned perception) is a function of vehicle uncertainty and can be shown in Figure 3-1.

As successive observations are made, landmark estimates become fully correlated [17]. This is the second convergence theorem proved by Newman in his thesis [38]. This implies that features become progressively more correlated as successive observations are made. This theorem also implies that in the limit, the map becomes fully correlated and given the exact location of any one landmark, the exact absolute location of the vehicle or any other feature is deduced [22]. Thus, as successive observations are made the error associated with vehicle estimates decrease monotonically [17]. These theorems motivate our approach.

When the uncertainty of the vehicle estimates grows, re-observing a feature already mapped will update and improve the state estimates. This is because the total uncertainty of the state estimate does not increase during an update [17]. The question then arises, which feature should the vehicle attempt to re-observe. Our method of determining the feature the vehicle heads toward is defined below.

This is the goal of our algorithm; in order to minimize vehicle uncertainty, planned perception determines which feature the vehicle should steer toward. Re-observing features will diminish vehicle uncertainty. Once vehicle uncertainty decreases and the determinant of $\mathbf{P}_{vv}$ is again below a given threshold, the vehicle returns to *exploration* and carries out its mission.

3.4.2 Criteria

The proposed planned perception approach considers three questions:

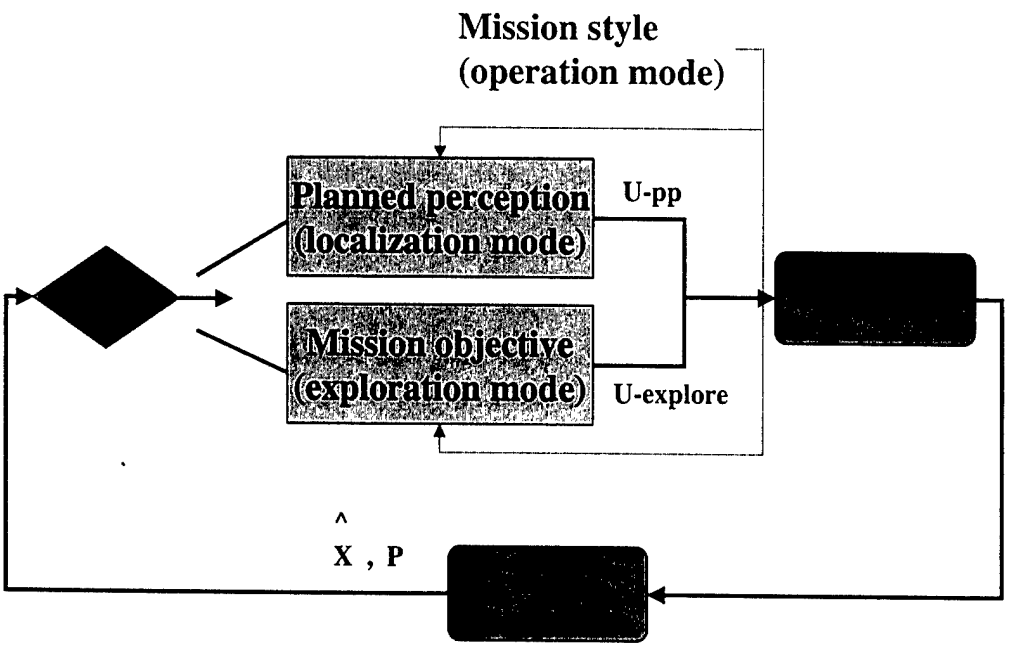


Figure 3-1: The Overall Mission Control Loop

1. For all mapped features $i = 1, \ldots, N_f$, the re-observation of which feature i will best improve vehicle estimates?

2. What is the cost of re-observing feature i ?

3. What is the risk of re-observing feature i ?

The first question relates to the estimate of the vehicle state and associated covariance matrix. The second question presents a means to account for the cost of moving toward feature i. Battery life and power are important to autonomous robots. Therefore, moving to a far away feature will require more energy to be spent. Feder states: the problem of associating measurements with features in cluttered environments remains a challenging and important problem for any estimation problem that faces data association ambiguities [21]. The third question address this problem of data association.

Question 1: For all mapped features $i = 1, \ldots, N_f$, the re-observation of which feature i will best improve vehicle estimates?

The determinant of $\mathbf{P}_{vv}$ is the measure of the volume of uncertainty associated with the vehicle state estimate. The total uncertainty of the state estimate does not increase during an update. An observation $\mathbf{z}_i(k+1)$ is used to update the state and covariance predictions yielding new estimates at time $k+1$. The new state update $\hat{\mathbf{x}}(k+1|k+1)$ and updated state covariance $\mathbf{P}(k+1|k+1)$ are computed

$$\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{W}(k+1)\nu(k+1), \tag{3.5}$$

$$\mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{W}(k+1)\mathbf{S}(k+1)\mathbf{W}^T(k+1), \tag{3.6}$$

where $\mathbf{W}$ is the weighted filter gain and $\mathbf{S}$ is the innovation covariance as described in Section 2.4. Thus, the covariance matrix is updated based on the matrix $\mathbf{W}(k+1)\mathbf{S}(k+1)\mathbf{W}^T(k+1)$. The matrix $\mathbf{WSW}^T$ can be represented as

$$\mathbf{WSW}^T = \left[\begin{array}{c|c} (\mathbf{WSW}^T)_{vv} & (\mathbf{WSW}^T)_{vf} \\ \hline (\mathbf{WSW}^T)_{vf}^T & (\mathbf{WSW}^T)_{ff} \end{array}\right]. \tag{3.7}$$

Since our approach focuses on minimizing vehicle uncertainty, we will concentrate on the sub-matrix $(\mathbf{WSW}^T)_{vv}$ which will be referred to as $\mathbf{WSW}_v^T$. The sub-matrix $\mathbf{P}_{vv}$ is updated by the sub-matrix $\mathbf{WSW}_v^T$ through the calculation

$$\mathbf{P}_{vv}(k+1|k+1) = \mathbf{P}_{vv}(k+1|k) - [\mathbf{W}(k+1)\mathbf{S}(k+1)\mathbf{W}(k+1)^T]_v \tag{3.8}$$

Thus, since we desire to maximize vehicle certainty, we want the maximum change $\mathbf{WSW}_v^T$ can provide. Representing the update Equation 3.8 in terms of determinants

$$|\ \mathbf{P}_{vv}(k+1|k+1)\ | = |\ \mathbf{P}_{vv}(k+1|k)\ | - |\ \mathbf{WSW}_v^T\ |, \tag{3.9}$$

it is clear that to minimize the updated covariance of the vehicle, we need to maximize $\mid \mathbf{WSW}_v^T \mid$.

While performing planned perception, the vehicle desires re-observe a known feature. Thus, the vehicle simulates a measurement to every known feature i. The associated, simulated matrix $(\mathbf{W}_i\mathbf{S}_i\mathbf{W}_i^T)_v$ is calculated for each feature. Because $\mid \mathbf{WSW}_v^T \mid$ needs to be maximized, the vehicle chooses to steer toward feature i based on the metric

$$arg_i\, max \mid (\mathbf{W}_i\mathbf{S}_i\mathbf{W}_i^T)_v \mid \tag{3.10}$$

The goal is to steer towards feature i that, if observed, would most reduce $\mathbf{P}_{vv}$.

Question 2: What is the cost of re-observing feature i ?

Energy efficiency is a motivator for choosing actions to be performed by autonomous robots. This question address the expense of driving the vehicle toward a relatively far landmark. The cost of determining to steer toward a distant feature is proportional to the distance between feature i and the robot's current position. In 2-Dimensional space, the distance (represented $\| \cdot \|$) between two points ($\mathbf{p}_1$ and $\mathbf{p}_2$) is calculated

$$\|\mathbf{p}_1 - \mathbf{p}_2\| = \sqrt{((\mathbf{p}_{1x}) - (\mathbf{p}_{2x}))^2 + ((\mathbf{p}_{1y}) - (\mathbf{p}_{2y}))^2} \tag{3.11}$$

where $\mathbf{p}_1 = [\mathbf{p}_{1x}\ \mathbf{p}_{1y}]^T$ and $\mathbf{p}_2 = [\mathbf{p}_{2x}\ \mathbf{p}_{2y}]^T$.

The distance between the current estimated robot position and feature i, $\|\hat{\mathbf{x}}_{fi} - \hat{\mathbf{x}}_v\|$, can be calculated according to Equation 3.11

$$\|\hat{\mathbf{x}}_{fi} - \hat{\mathbf{x}}_v\| = \sqrt{(\mathbf{x}_{f_i} - x_v)^2 + (\mathbf{y}_{f_i} - y_v)^2} \tag{3.12}$$

This distance $\|\hat{\mathbf{x}}_{fi} - \hat{\mathbf{x}}_v\|$ serves as our second criteria to determining the feature the vehicle steers toward.

Question 3: What is the risk of re-observing feature i?

This question is a one of data association. Cluttered environments pose a threat to correctly associating measurements with features. Our approach considers the consequences of data association errors present in cluttered environments. For each estimated feature $i = 1, \ldots, N_f$, the mean number of features within a certain distance are calculated. This distance is a defined threshold (i.e. two meters) and is used to calculate the density of features in a certain area. If feature density is relatively large, a measurement may be associated with the wrong feature; this may cause divergence due to a data association error [21].

Thus, our third and final criteria for determining which feature the vehicle should steer toward is a function of the density of features surrounding feature i, termed $density_{fi}$. The mean number of features within a certain distance is calculated to serve as the density function.

3.4.3 Algorithm

As described in Section 3.4.2, there are three aspects which govern our planned perception algorithm of choosing which feature to steer toward. The criteria are summarized

- Which feature, when re-observed, will best reduce vehicle uncertainty?
- What is the cost of driving toward a feature?
- What is the risk of associating a measurement of that feature with the wrong feature?

We want to develop a quantitative method for evaluating which feature the vehicle should head toward. Combining these three criteria into one equation involves $\mid (\mathbf{W}_i\mathbf{S}_i\mathbf{W}_i^T)_v \mid$, $\|\hat{\mathbf{x}}_{fi} - \hat{\mathbf{x}}_v\|$, and a function of the density of features surrounding feature i, $f(density_{fi})$. The three criteria will be represented as a scalar number, Γ, calculated

$$\Gamma_i = -\alpha \mid (\mathbf{W}_i\mathbf{S}_i\mathbf{W}_i^T)_v \mid + \beta\|\hat{\mathbf{x}}_{fi} - \hat{\mathbf{x}}_v\| + \zeta \cdot f(density_{fi}) \tag{3.13}$$

where α, β, and ζ are weighted gains for the different criteria. α represents the gain for the information desired. β represents the gain for the distance between the robot and feature i. ζ is the gain for the measure of the density of landmarks around feature i. Our goal is to minimize $\mathbf{\Gamma}_i$ in Equation 3.13.

While the determinant of $\mathbf{P}_{vv}$ is below the given threshold, the vehicle performs according to its assigned mission; while $\mathbf{P}_{vv}$ is relatively small, explore. When the vehicle is no longer "confident" in its state estimate, the planned perception algorithm acts as the controller and determines which feature to steer toward. This can be seen in Figure 3-1.

The *localization mode*, that of adaptively sensing the environment through planned perception, is based on Equation 3.13. This equation is computed for all mapped features by simulating a measurement to each feature. Each mapped feature, $i = 1, \ldots, N_f$, has an associated $\mathbf{\Gamma}_i$ value. When all $\mathbf{\Gamma}_i$ are computed for one iteration of the filter, the vehicle's control inputs are changed to steer toward feature $i*$ that is associated with the minimum value $\mathbf{\Lambda}_{i*}$ defined

$$\mathbf{\Lambda}_{i*} = arg_i \; min \; \mathbf{\Gamma}_i. \tag{3.14}$$

Thus, the vehicle chooses to steer toward the feature $i*$ associated with $\mathbf{\Lambda}_{i*}$ in Equation 3.14.

Performing the above planned perception approach will cause the vehicle to possibly desire to steer toward a different feature $i*$ at each time step. This continuous changing of desired features may cause the vehicle to oscillate between control inputs. To compensate for this oscillation, we add in some hysterisis.

Hysterisis is a retardation of the effect of an action. Adding some hysterisis into our decision process accounts for the possibility that our algorithm may determine the vehicle should head toward a different feature $i*$ at each iteration. Therefore, to ensure that our vehicle is steering toward the feature which bests suits our criteria for the entire time it is in localization mode, we store the value $\mathbf{\Lambda}_{i*}$ at each time step. This provides us with $\mathbf{\Lambda}_{i*}(k)$.

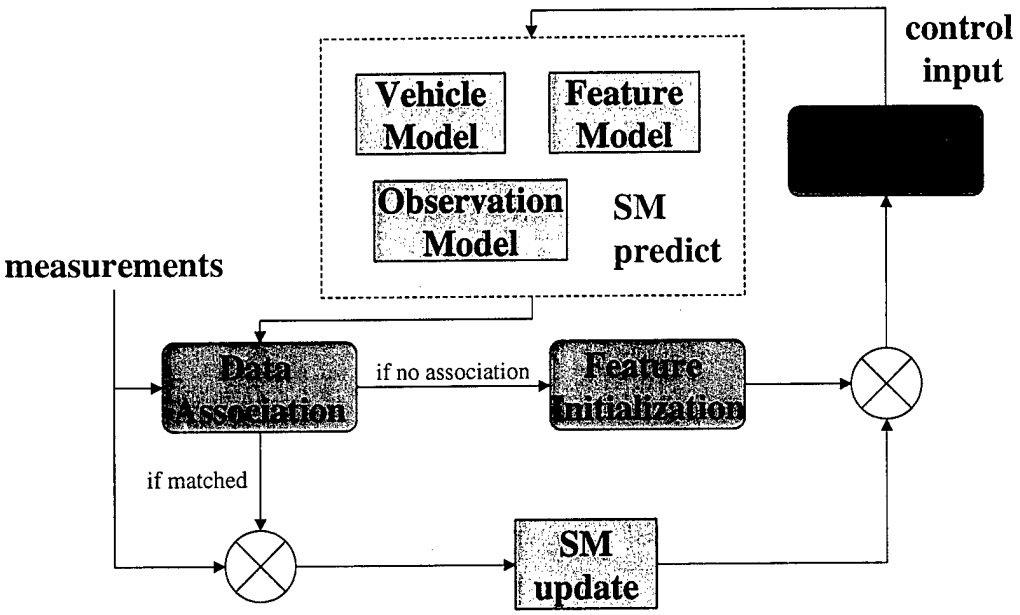


Figure 3-2: The integration of Planned Perception in Stochastic Mapping

At each time step k during the localization mode of mission, an associated $\mathbf{\Lambda}_{i*}(k)$ value is stored. This value is compared to $\mathbf{\Lambda}_{i*}(k-1)$. If

$$\mathbf{\Lambda}_{i*}(k) \leq \mathbf{\Lambda}_{i*}(k-1) \tag{3.15}$$

the new feature $i*(k)$ associated with $\mathbf{\Lambda}_{i*}(k)$ is the feature the vehicle determines to steer toward. However, if $\mathbf{\Lambda}_{i*}(k) > \mathbf{\Lambda}_{i*}(k-1)$, the vehicle continues to steer toward the feature associated with $\mathbf{\Lambda}_{i*}(k-1)$.

The action of the robot is given by evaluating Equation 3.14 using the metric Equation 3.15. The vehicle steers towards the feature which satisfies the above criteria.

3.4.4 Planned perception summary

Planned perception is integrated into stochastic mapping as shown in Figure 3-2 where the planned perception step is derived from Figure 3-1. The algorithm for performing planned perception is summarized below.

1. **while** $\det(\mathbf{P}_{vv}) \leq \det(\mathbf{P}_{vvthreshold})$ **do** $\longrightarrow$ **explore**
2. **control** steer toward current waypoint
3. if distance to current waypoint is < 3 m, **control** steer toward next waypoint
4. end **while**
5. **while** $\det(\mathbf{P}_{vv}) > \det(\mathbf{P}_{vvthreshold})$ **do** $\longrightarrow$ **planned perception**
6. $\forall$ features $i = 1, \ldots, N_f$, simulate measurement
7. **for** each $feature_i$ and simulated measurement, **calculate**
8. Innovation Covariance $\mathbf{S}$
9. Filter Gain $\mathbf{W}$
10. $\mathbf{W}_i\mathbf{S}_i\mathbf{W}_i^T$
11. $\det(\mathbf{W}_i\mathbf{S}_i\mathbf{W}_i^T)$
12. $\|\hat{\mathbf{x}}_{fi} - \hat{\mathbf{x}}_v\|$
13. mean number of features around $feature_i \longmapsto density_{fi}$
14. $\Gamma_i = -\alpha \det(\mathbf{W}_i\mathbf{S}_i\mathbf{W}_i^T) + \beta\|\hat{\mathbf{x}}_{fi} - \hat{\mathbf{x}}_v\| + \zeta(density_{fi})$
15. $\mathbf{\Lambda}_{i*}(k) = arg_i\ min\ \Gamma_i(k)$
16. if $\mathbf{\Lambda}_{i*}(k) \leq \mathbf{\Lambda}_{i*}(k-1) \longmapsto$ store $\mathbf{\Lambda}_{i*}(k)$, otherwise $\mathbf{\Lambda}_{i*}(k-1) = \mathbf{\Lambda}_{i*}(k)$
17. Vehicle heads toward feature $i* \longrightarrow$ **control** steer to **feature** $i*$
18. end **while**

3.5 Summary

This chapter presented the theory and algorithm of our approach to performing planned perception. The idea is based on analyzing the uncertainty in vehicle estimates. Planned perception determines what feature the vehicle should head toward. The presented strategy of performing adaptive sensing can easily be incorporated into the stochastic mapping algorithm.

Chapter 4

Simulation Design and Results

This chapter is concerned with the simulation design and results. A description of the simulator is presented in Section 4.1. This chapter also presents the results from the simulations of the integrated planned perception and concurrent mapping and localization (CML) algorithm described in Chapter 2 and Chapter 3.

4.1 Simulation design

This section describes the simulation designed to implement the incorporation of planned perception within CML. The simulation is coded in © MATLAB and is described in Figure 4-1. Each step of the simulation is presented below. This presentation is a general overview of the simulation based on the algorithm and procedures set forth in Chapter 2 and Chapter 3. For a more in depth discussion on each process in Figure 4-1, refer to these Chapters.

4.1.1 Setup and initialization

This step of the simulation process is performed as the first step of the simulation. It defines and initializes all variables that will be used throughout the estimation process. During the setup, all matrices, vectors, and variables are defined and the specific size of each matrix is set. The estimated state vector, $\hat{\mathbf{x}}$, is initialized to

zero with an initial uncertainty. Vehicle uncertainty is initialized with a standard deviation of 0.1 m in both the x and y direction, and a heading standard deviation of 0.1 degree.

The setup process also defines the parameters of the system. The parameters are defined in Table 4.1.

Table 4.1: System parameters

sampling period, ΔT,	1 s
vehicle cruise speed, v	.75 m/s
max. steer angle of vehicle	22.5^o
speed process noise std. dev.	0.25 m/s
heading process noise std. dev.	1.0^o
range of sensor	25 m
sensor field of view	$\pm 90^o$
probability of measurement return	90%
sensor range noise std. dev.	0.1 m/s
sensor bearing noise std. dev.	1^o

The noise inherent in the process and observation models are also defined. The noise covariances, $\mathbf{Q}$ and $\mathbf{R}$, are associated with the vehicle and observation models, respectively. The covariances are discussed in Section 2.3.1 and Section 2.3.3 and are defined

$$\mathbf{Q} = \begin{bmatrix} 0.5^2 & 0 \\ 0 & 1^2 \end{bmatrix} \tag{4.1}$$

$$\mathbf{R} = \begin{bmatrix} 0.1^2 & 0 \\ 0 & 1^2 \end{bmatrix}. \tag{4.2}$$

The parameter and noise values are not chosen arbitrarily. They are chosen to reflect actual values apparent in systems. The sensor is modelled after the SICK laser scanner used in the Marine Robotics Laboratory at the Massachusetts Institute of Technology. The vehicle is modelled to have two scanners to provide range and bearing measurements. The range, field of view, and probability to return a measurement are described in Table 4.1.

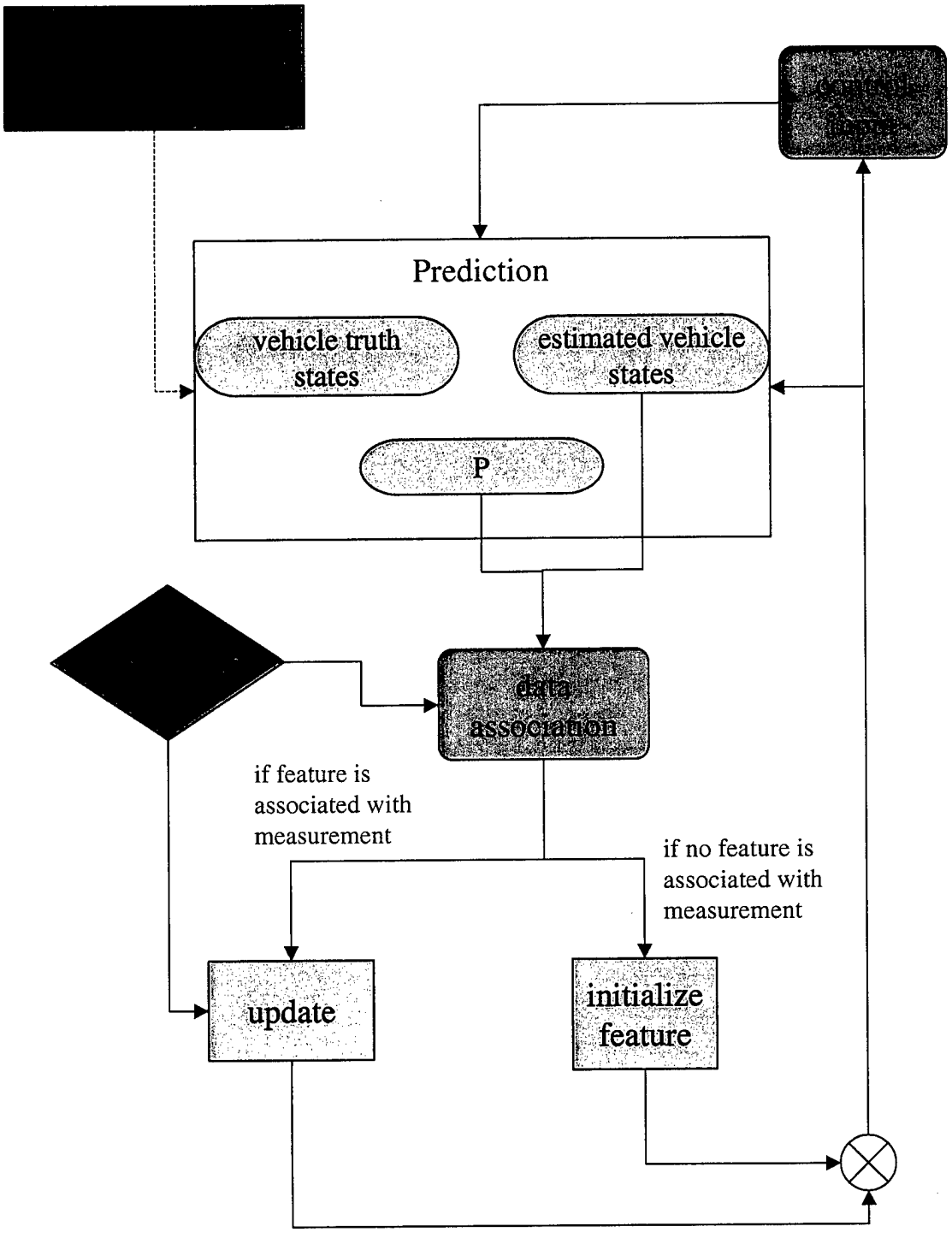


Figure 4-1: The Simulation Design.

The parameters and defined matrices are passed to the filter.

4.1.2 Estimation process

This section describes the estimation process of the simulation. The estimation process consists of the prediction, data association, update, and feature initialization steps of the simulation.

Prediction

The prediction step of stochastic mapping described in Section 2.4.1. It consists of predicting the true vehicle states, the estimated vehicle states, and the covariance matrix.

The true and estimated vehicle states are predicted by propagating the previous states through the vehicle model. The states of the vehicle are predicted by

$$\begin{bmatrix} x_v \\ y_v \\ \theta \end{bmatrix} |_{k+1} = \begin{bmatrix} x_v + v\Delta T\cos(\theta) \\ y_v + v\Delta T\sin(\theta) \\ \theta + \varphi \end{bmatrix} |_k, \tag{4.3}$$

where v and φ are speed and heading control inputs with the process noise given in Table 4.1. The control inputs are given from the control input step of the simulation. This process will be described in Section 4.1.4.

The covariance matrix $\mathbf{P}$ is also estimated during the prediction step. To estimate $\mathbf{P}$, the Jacobians $\nabla\mathbf{f}_v$ and $\nabla\mathbf{f}_u$ are calculated according to Equation 2.14 and Equation 2.28, respectively. The prediction of the covariance matrix is

$$\mathbf{P}(k+1|k) = \nabla\mathbf{f}_v(k)\mathbf{P}(k|k)\nabla\mathbf{f}_v{}^T(k) + \nabla\mathbf{f}_u(k)\mathbf{u}(k)\nabla\mathbf{f}_u{}^T(k) + \mathbf{Q}(k). \tag{4.4}$$

The prediction process is described as

1. **do** $\rightarrow$ **prediction**
2. $\mathbf{x}(k+1|k = \mathbf{f}_v[\mathbf{x}_v(k), \mathbf{u}_v(k+1)] \Leftrightarrow$ (truth projection)
3. $\hat{\mathbf{x}}_v(k+1) = \mathbf{f}_v[\hat{\mathbf{x}}_v(k), \mathbf{u}_v(k+1)] \Leftrightarrow$ (estimate projection)
4. $\mathbf{Q} \leftarrow \mathbf{u}(k) \Leftrightarrow$ (process noise covariance)
5. $\nabla \mathbf{f}_v \Leftrightarrow$ (Jacobian of w.r.t. vehicle state)
6. $\nabla \mathbf{f}_u \Leftrightarrow$ (Jacobian of w.r.t. control input)
7. $\mathbf{P}(k+1|k) = \nabla \mathbf{f}_v(k)\mathbf{P}(k|k)\nabla \mathbf{f}_v{}^T(k) + \nabla \mathbf{f}_u(k)\mathbf{u}(k)\nabla \mathbf{f}_u{}^T(k) + \mathbf{Q}(k) \Leftrightarrow$ (projection)
8. **end prediction**

The predicted estimated state vector and covariance matrix are used to attempt to assign features to measurements received from the sensors in the data association step of the simulation.

Data association

Data association is relating observations and measurements to their corresponding features. Data association algorithms strive to properly assign measurements to the features from which they originate. Nearest neighbor gating, the process described in Section 2.4.4, is the data association method simulated.

Nearest neighbor gating is simulated as

1. **do** $\rightarrow$ **data association**
2. **for** $\forall$ features, $i = 1 \ldots N_f$, **calculate**
3. $\nu_i = \mathbf{z}_i - \hat{\mathbf{z}}_i \Leftrightarrow$ (innovation)
4. $\nabla \mathbf{h}_x \Leftrightarrow$ (Jacobian of $\mathbf{h}$ w.r.t. state estimate)
5. $\mathbf{S}_i(k+1) = \nabla \mathbf{h}_x(k)\mathbf{P}(k+1|k)\nabla \mathbf{h}_x{}^T(k) + \mathbf{R}(k+1) \Leftrightarrow$ (innovation covariance)
6. $\gamma_i = \nu_i^T \mathbf{S}_i^{-1} \nu_i \leq 9 \Leftrightarrow$ (association test)
7. if $\gamma_i \leq 9 \rightarrow$ **update**
8. **end for**
9. if no feature gates, $\rightarrow$ **do** $\rightarrow$ **feature initialization**
10. **end data association**

The gating is performed where 9 is chosen as the gating parameter based on a χ^2 distribution as defined in Equation 2.40. Also, based on the estimated state vector

and observation model described in Section 2.3.3, the Jacobian $\nabla \mathbf{h}_x$ is calculated according to Equation 2.33.

Update

The update step of simulation is straightforward; this step updates the estimates of the state vector and covariance matrix based on the observation $\mathbf{z}$. The estimates are updated, as described in Section 2.4.3, according to

1. **do** $\rightarrow$ **update**
2. $\quad \mathbf{W}(k+1) = \mathbf{P}(k+1|k)\nabla \mathbf{h}_x{}^T(k+1)\mathbf{S}^{-1}(k+1) \Leftrightarrow$ (weighted gain / Kalman gain)
3. $\quad \hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{W}(k+1)\nu(k+1) \Leftrightarrow$ (state update)
4. $\quad \mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{W}(k+1)\mathbf{S}(k+1)\mathbf{W}^T(k+1) \Leftrightarrow$ (covariance update)
5. **end update**

The updated estimates are passed to the control input step to determine how the vehicle should act. The choice of control input is described in Section 4.1.4.

Feature initialization

This step of the simulation is called when no previously mapped feature is associated with a measurement. Feature initialization augments the estimated state vector and covariance to incorporate the new feature's location and uncertainty. The augmentation is simulated as

1. **do** → **feature initialization**
2. $\hat{\mathbf{x}}(k+1|k+1) = \left[\begin{array}{c} \hat{\mathbf{x}}(k+1|k) \\ \hline \hat{\mathbf{x}}_{f_{N_f+1}} \end{array}\right]$ ⇔ (augmented state estimate)
3. $\mathbf{B} = \nabla \mathbf{g}_x \mathbf{P}(k+1|k)$
4. $\mathbf{A} = \nabla \mathbf{g}_x \mathbf{P}(k+1|k) \nabla \mathbf{g}_x{}^T + \nabla \mathbf{g}_z \mathbf{R} \nabla \mathbf{g}_z{}^T$
5. $\mathbf{P}(k+1|k+1) = \left[\begin{array}{c|c} \mathbf{P}(k+1|k) & \mathbf{B}^T \\ \hline \mathbf{B} & \mathbf{A} \end{array}\right]$ ⇔(augmented covariance matrix)
6. increase number of features
7. **end feature initialization**

This step of the simulation calculates the appropriate Jacobians to initialize the feature as described in Section 2.4.5.

4.1.3 Measurements

Measurements are simulated as ranges and bearings to features. Ranges and bearings are simulated as

1. **do** → **measurement**
2. **for** ∀ features, $i = 1 \ldots N_f$
3. **if** feature i is in sensor's field of view, add feature position to temporary set
4. from this set, **randomly** choose **1 feature**⇔(feature observing)
5. generate **random number** between 0 and 1
6. **if** random number $> 0.9 \rightarrow \mathbf{Z} = [\,]$ ⇔(false return)
7. **else**, $\mathbf{Z} = \left[\begin{array}{c} \sqrt{(\mathbf{x}_{f_i}(k) - x_v(k))^2 + (\mathbf{y}_{f_i}(k) - y_v(k))^2} \\ \hline \arctan \frac{\mathbf{y}_{f_i}(k) - y_v(k)}{\mathbf{x}_{f_i}(k) - x_v(k)} - \theta_v \end{array}\right]$
8. **if** no feature is in the sensor's field of view, $\mathbf{Z} = [\,]$ ⇔(no return)
9. **end for**
10. **end measurement**

The measurement process described above is simulated for every time step.

4.1.4 Control input

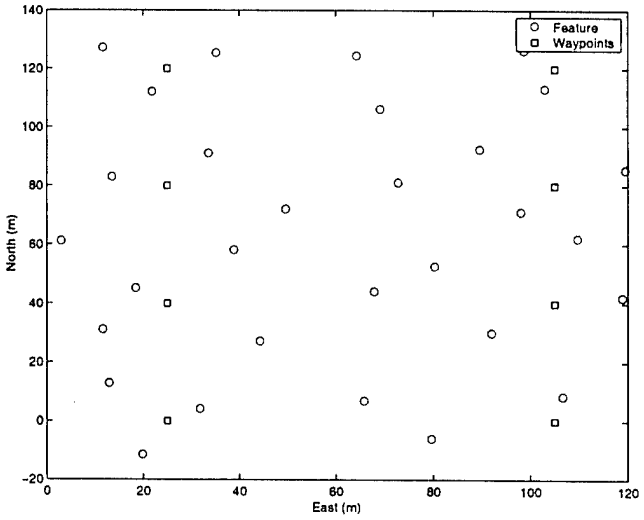
This step of the simulation determines the control input to the vehicle. The control input is a function of the uncertainty in the vehicle, defined $\mathbf{P}_{vv}$. While the uncertainty is less than the threshold, $\mathbf{P}_{vvthreshold}$, the vehicle moves from waypoint to waypoint. When the vehicle is within 3 meters of the current waypoint, the control heading is changed to steer the vehicle to the next waypoint. While the uncertainty of the vehicle is greater than the given threshold, the vehicle performs planned perception. The planned perception algorithm is described in Chapter 3.

The process of switching between mission styles, exploration and planned perception, is simulated

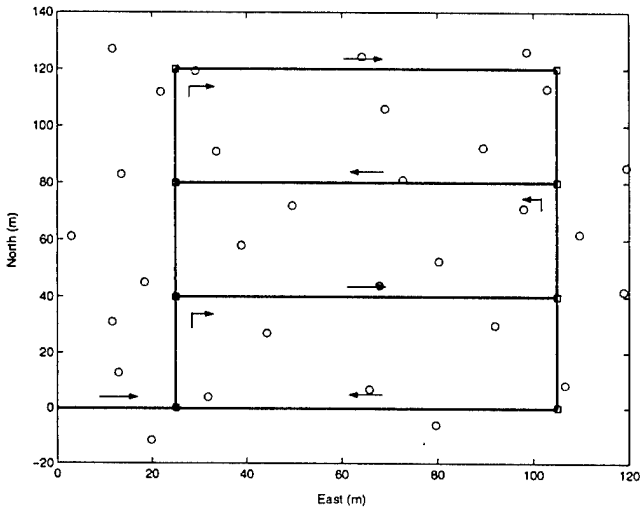
1. **while** $\det(\mathbf{P}_{vv}) \leq \det(\mathbf{P}_{vvthreshold})$ **do** $\longrightarrow$ **explore**
2. **control** steer toward current waypoint
3. if distance to current waypoint is < 3 m, **control** steer toward next waypoint
4. end **while**
5. **while** $\det(\mathbf{P}_{vv}) > \det(\mathbf{P}_{vvthreshold})$ **do** $\longrightarrow$ **planned perception**
6. $\forall$ features $i = 1, \ldots, N_f$, simulate measurement
7. **for** each $feature_i$ and simulated measurement, **calculate**
8. Innovation Covariance $\mathbf{S}$
9. Filter Gain $\mathbf{W}$
10. $\mathbf{W}_i\mathbf{S}_i\mathbf{W}_i^T$
11. $\det(\mathbf{W}_i\mathbf{S}_i\mathbf{W}_i^T)$
12. $\|\hat{\mathbf{x}}_{fi} - \hat{\mathbf{x}}_v\|$
13. mean number of features around $feature_i \longmapsto density_{fi}$
14. $\Gamma_i = -\alpha \det(\mathbf{W}_i\mathbf{S}_i\mathbf{W}_i^T) + \beta\|\hat{\mathbf{x}}_{fi} - \hat{\mathbf{x}}_v\| + \zeta(density_{fi})$
15. $\mathbf{\Lambda}_{i*}(k) = arg_i\ min\ \Gamma_i(k)$
16. if $\mathbf{\Lambda}_{i*}(k) \leq \mathbf{\Lambda}_{i*}(k-1) \longmapsto$ store $\mathbf{\Lambda}_{i*}(k)$, otherwise $\mathbf{\Lambda}_{i*}(k-1) = \mathbf{\Lambda}_{i*}(k)$
17. Vehicle heads toward feature $i* \longrightarrow$ **control** steer to **feature** $i*$
18. end **while**

4.2 Simulation Results

In this simulation, the vehicle is to survey an area steering toward waypoints. The vehicle starts at position (0,0). The waypoints are defined to provide a "lawnmower" pattern, survey of the environment. The locations of the features and waypoints are defined during the setup process of the simulation in Figure 4-1. The features are 2-Dimensional point features and are chosen from the set where x and y are random, independent variables which are uniformly distributed over the area [x:0→120] and [y:-20→135]. The features, and waypoints define the environment and are shown in the top of Figure 4-2. The desired path of the vehicle is shown in the bottom of Figure 4-2.



Environment description



Desired Path of Vehicle

Figure 4-2: **Top:** The simulation environment. The red circles are the feature locations. The black squares are the waypoints. **Bottom:** The desired path of the vehicle navigating between waypoints is shown by the thick red line. The black arrows depict the direction the vehicle takes going from (0,0) to the waypoint (105,120). The vehicle then back-tracks performing the lawnmower survey.

In the first simulation, the vehicle moves between the waypoints navigating only by dead reckoning. In the second simulation, the vehicle performs CML as it navigates according to the stochastic mapping algorithm defined in Chapter 2. The third simulation integrates planned perception and CML. This simulation is performed constraining the overall volume of vehicle uncertainty (the determinant of $\mathbf{P}_{vv}$). Finally, the fourth simulation performs planned perception constraining the vehicle uncertainty in the x and y directions. This simulation constrains a subset of the determinant $\mathbf{P}_{vv}$; it constrains the determinant of $\mathbf{P}_{xy}$ which is the area of vehicle uncertainty associated with the x and y positions estimates. The simulations integrating planned perception and CML are broken into scenarios representing different strategies for planned perception navigation.

The global parameters for each simulation are defined in Table 4.1 and Table 4.2.

Table 4.2: Simulation parameters

sampling period, ΔT,	1 s
duration of simulation	2000 s
number of waypoints	12
number of features	30
initial vehicle x position uncertainty std. dev	0.1 m
initial vehicle y position uncertainty std. dev	0.1 m
initial vehicle heading uncertainty std. dev	0.1 deg

4.2.1 Dead reckoning

Navigating by dead reckoning means that the estimation is calculated only by the vehicle model. There is no observation of features. In terms of the Kalman filter based stochastic mapping algorithm, dead reckoning is navigating based on the prediction step alone. Dead reckoning is the best estimate of the vehicle model without the addition of noise. Thus, since the control input is known for the desired path, the estimated path navigating by dead reckoning is the same as the desired path. This can be seen in Figure 4-3.

Figure 4-3 shows the trajectory paths of the true vehicle and the estimated dead reckoning path. The absolute position error for dead reckoning grows with time.

This error grows without bounds as a result of the process noise. The errors in vehicle states are shown in Figure 4-4. Finally, the variances in x and y position are shown in Figure 4-5. The variances show how the position error grows unbounded.

Dead reckoning is navigating by projecting the vehicle through the vehicle model. There are no observations of features to "reset" the vehicle estimate. Thus, the error grows without bounds.

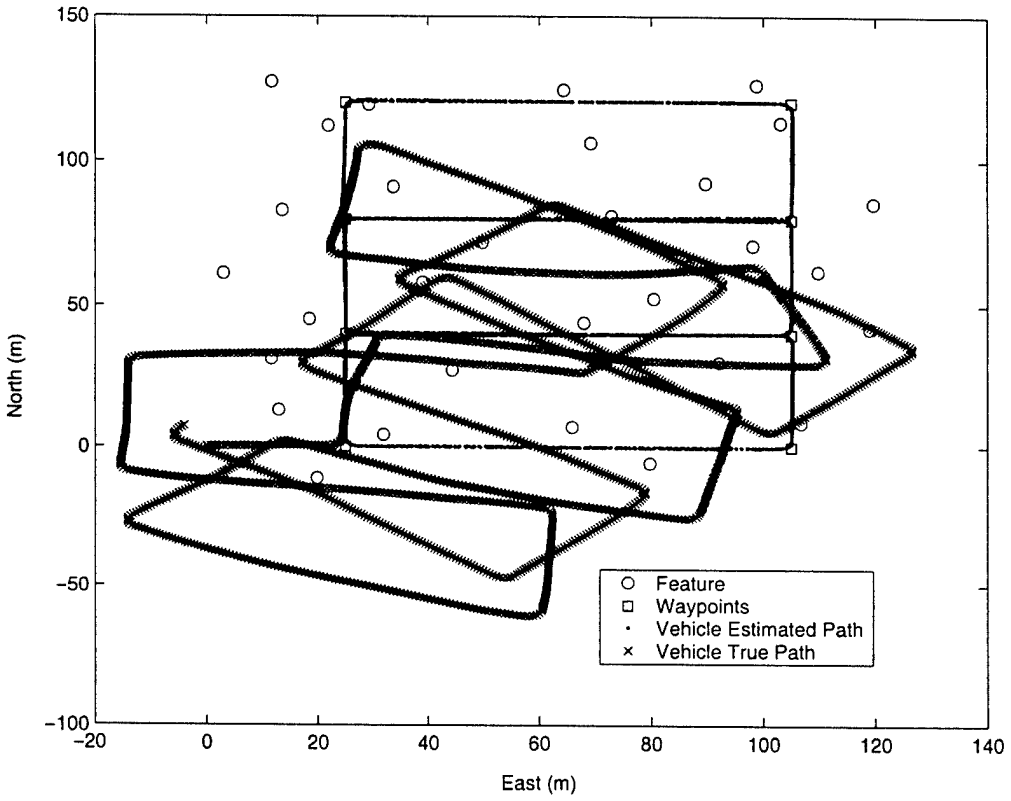


Figure 4-3: Dead Reckoning. True vehicle path versus estimated vehicle path. The red crosses represent the true vehicle path. The blue dots follow the estimated vehicle path. While performing dead reckoning, the estimated vehicle path is the same as the desired path. The true vehicle path differs because of the presence of process noise.

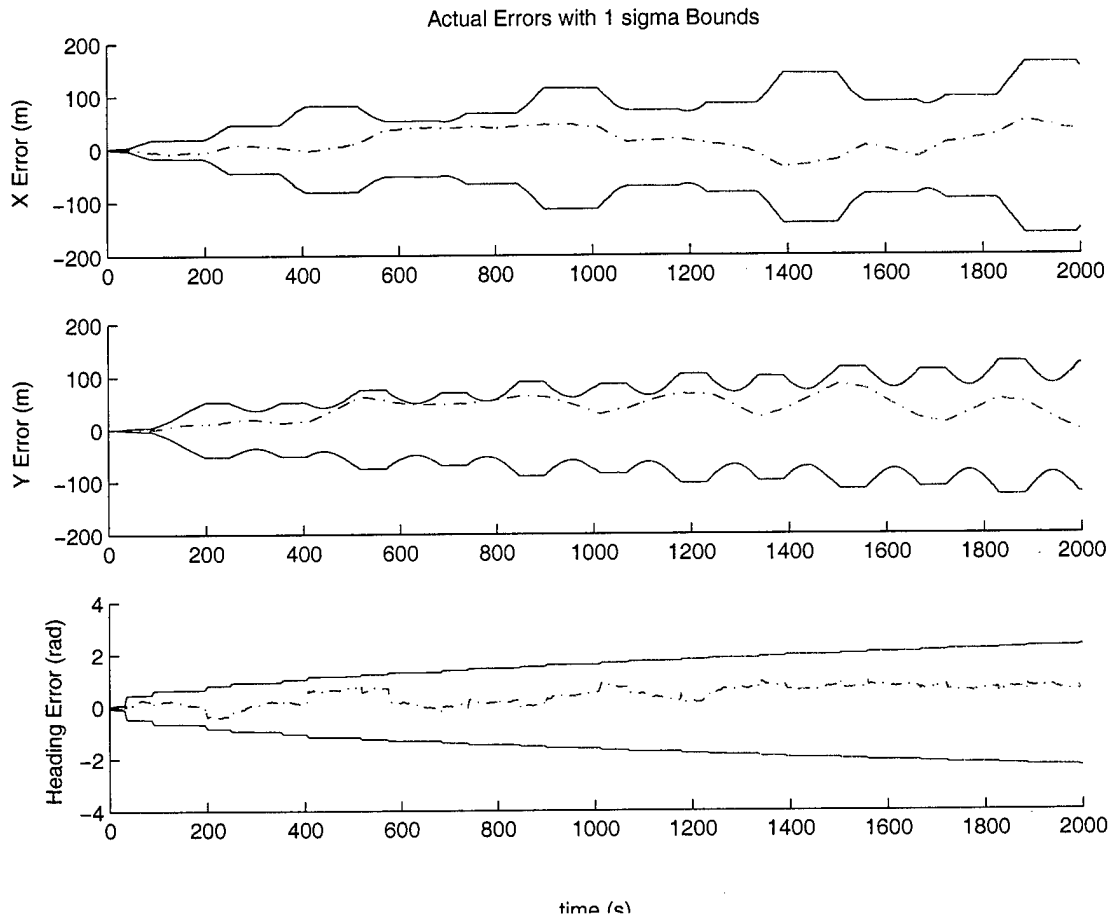


Figure 4-4: Dead Reckoning. The errors for the vehicle states when navigating by dead reckoning are shown. The actual errors are represented by the red dashed line. The solid blue lines represent $\pm 1\sigma$ error bounds. The errors grow without bounds.

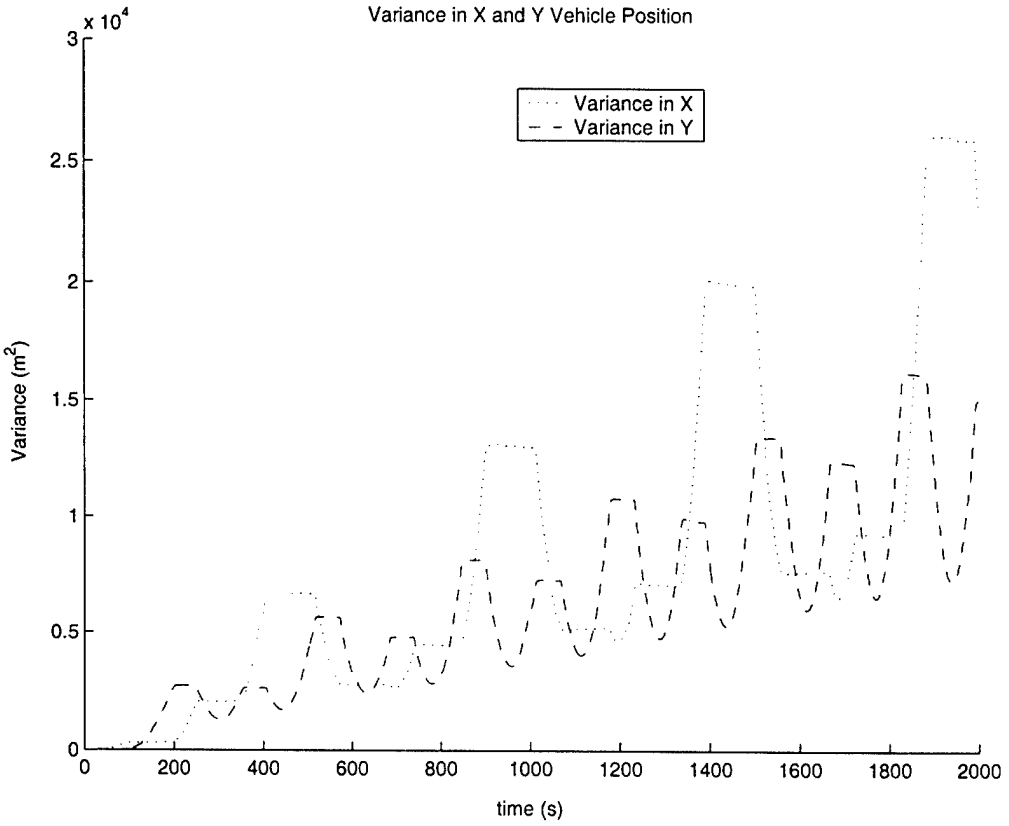
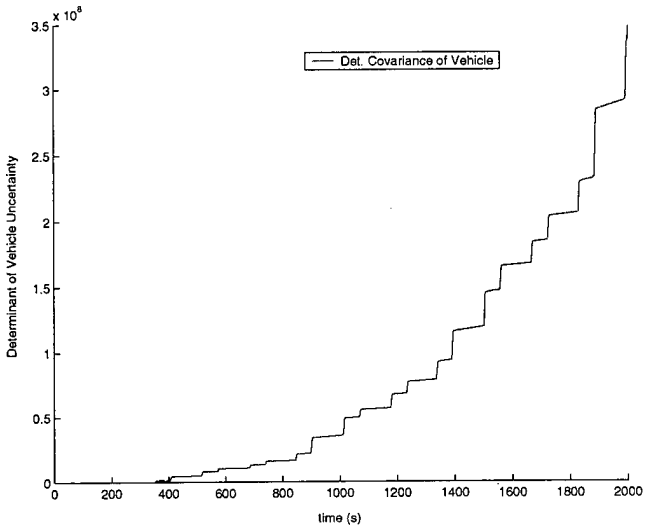
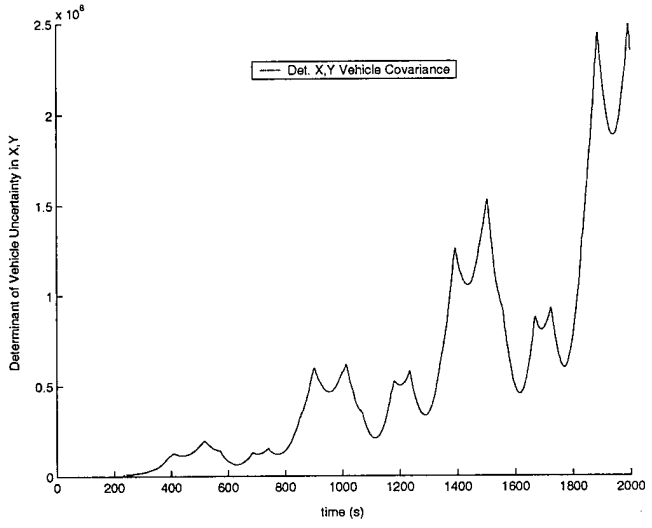


Figure 4-5: Dead Reckoning. The variances in x and y position are plotted. The red dotted line is the variance in x while the dashed blue line is the variance in y. As expected, these variances grow without bounds.



Determinant of Vehicle Uncertainty



Determinant of Vehicle x,y Uncertainty

Figure 4-6: Dead Reckoning. **Top:** The determinant of the overall vehicle uncertainty, $\mathbf{P}_{vv}$. **Bottom:** The determinant of the uncertainty in the vehicle x and y position, $\mathbf{P}_{xy}$.

4.2.2 Concurrent mapping and localization

This simulation navigates using the stochastic mapping algorithm defined in Chapter 2. Unlike dead reckoning, stochastic mapping utilizes observations of objects in the environment; stochastic mapping is a feature-based approach to CML. Observing features is used to obtain reference points for navigation. these features serve as localization points whose re-observation reduces vehicle uncertainty.

The estimated path and true path of the vehicle are compared in Figure 4-7. The CML error estimates are bounded as shown in Figure 4-8. They converge to lower limits as defined by the CML convergence theorems described in Section 2.5. The vehicle uncertainty converges to the initial vehicle uncertainty as successive observations are made. Because the estimated uncertainty can only be decreased during an update, the vehicle uncertainty can only be as good as the uncertainty that the vehicle possessed at the time it first observed a feature. The variances in the vehicle estimate converge as shown in Figure 4-9.

The volume of uncertainty in the vehicle can be represented as the determinant of the covariance matrix. The top of Figure 4-10 shows the determinant of the vehicle uncertainty, $\mathbf{P}_{vv}$. The x and y uncertainty is shown in the bottom of Figure 4-10 as the determinant of the uncertainty in $\mathbf{P}_{xy}$. The jumps in these figures represent the vehicle re-observing a feature and updating its estimate. This re-observation of features proves to lower the uncertainty in the vehicle estimate.

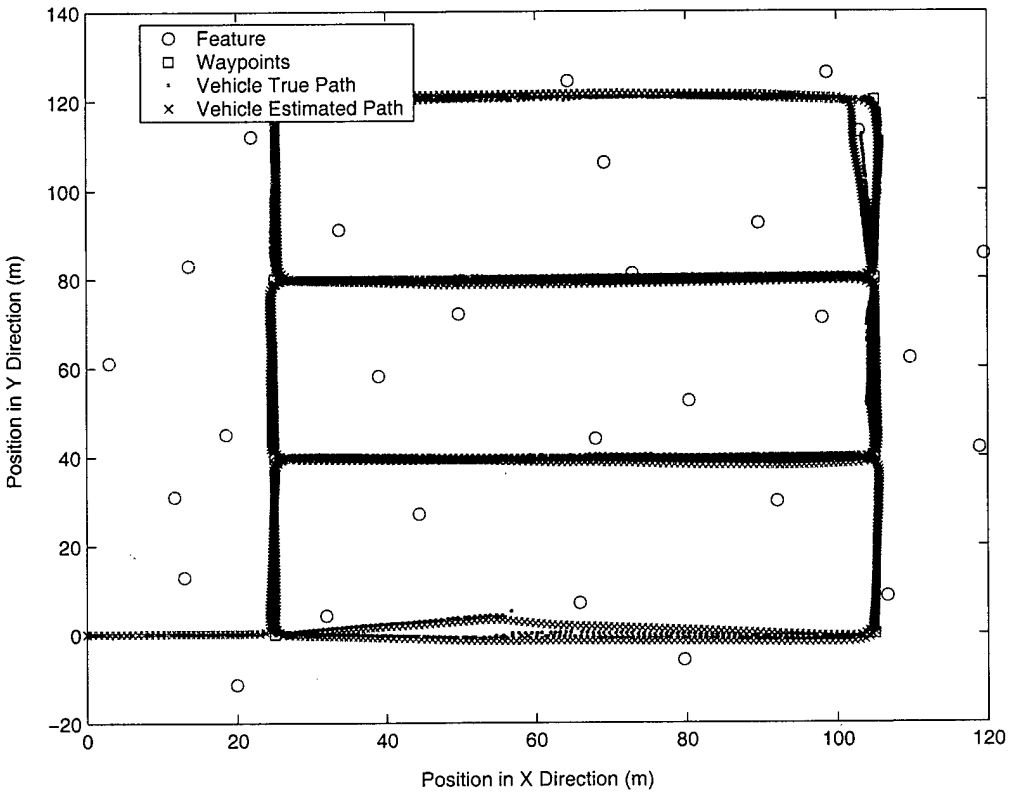


Figure 4-7: Concurrent Mapping and Localization. True vehicle path versus estimated vehicle path. The red crosses represent the true vehicle path. The blue dots follow the estimated vehicle path. The minor jumps in estimated and true position occur when the vehicle obtains an update and improves its estimated position.

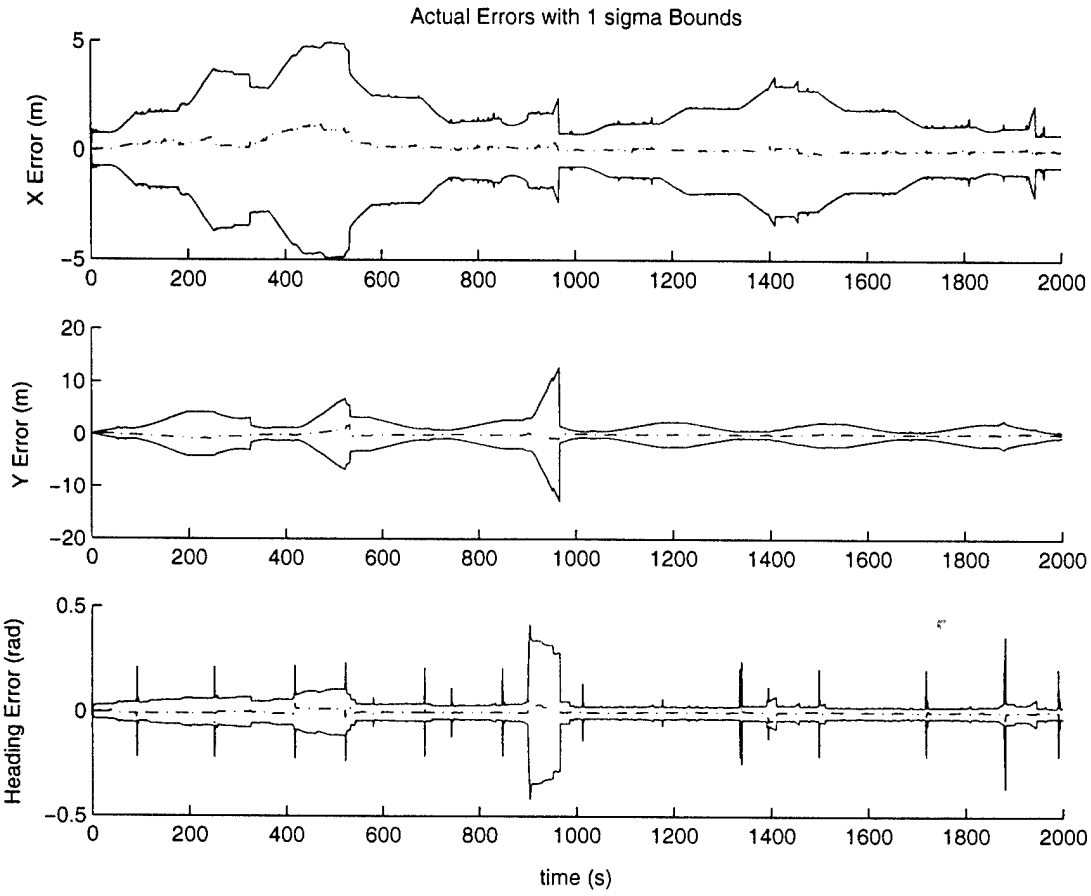


Figure 4-8: Concurrent Mapping and Localization. When performing CML, the errors are bounded. The actual errors are represented by the red dashed line. The solid blue lines represent $\pm 1\sigma$ error bounds. As expected, these errors converge.

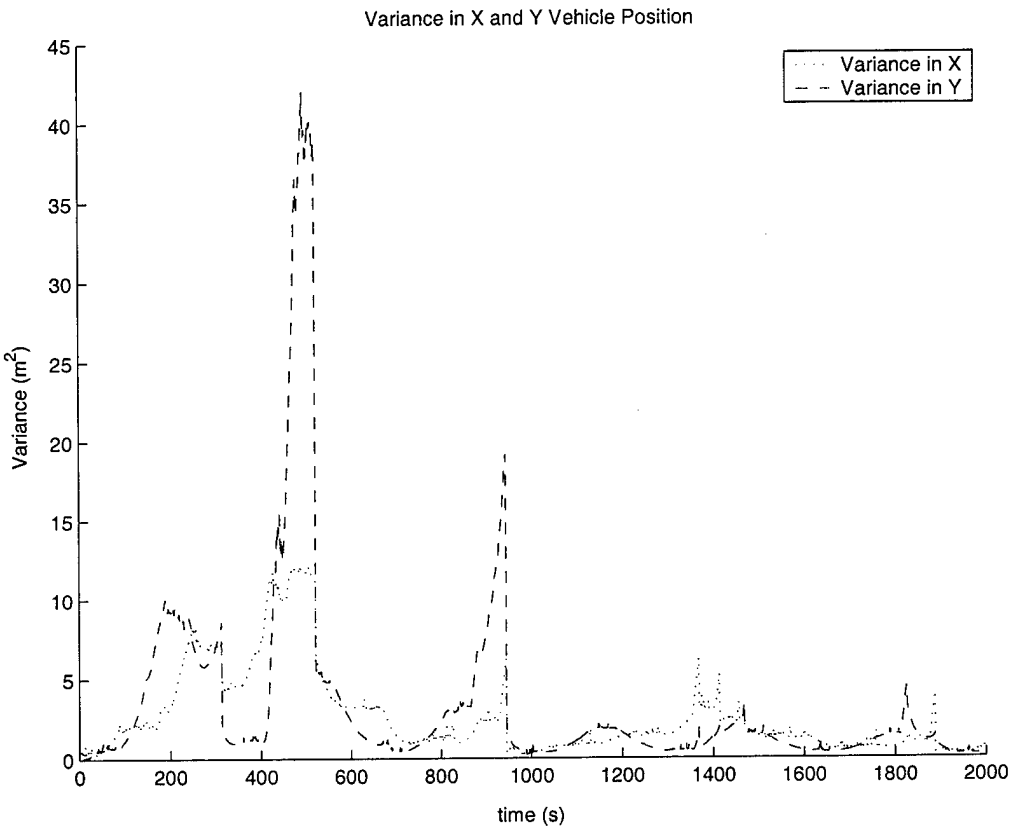
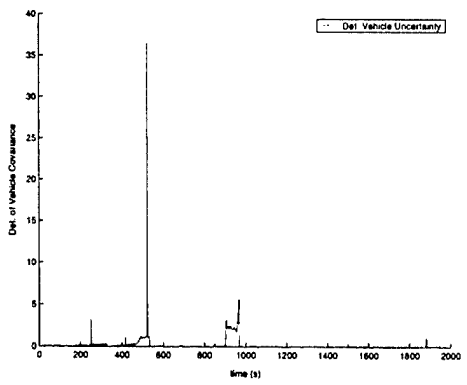
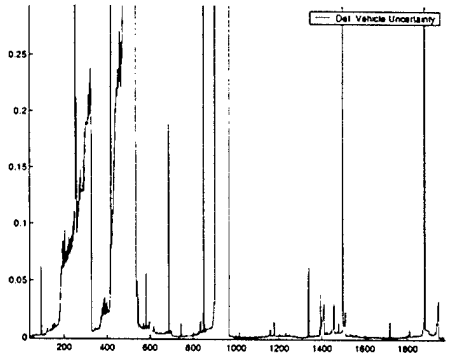


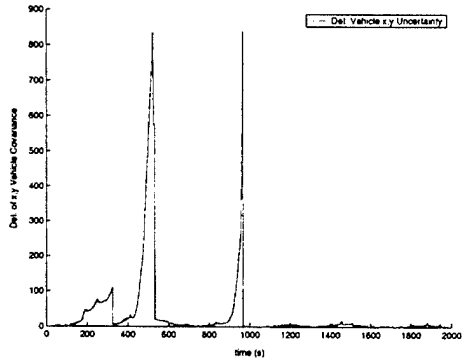
Figure 4-9: Concurrent Mapping and Localization. The variances in x and y position are plotted. The red dotted line is the variance in x while the dashed blue line is the variance in y. As expected, these variances converge to lower limits.



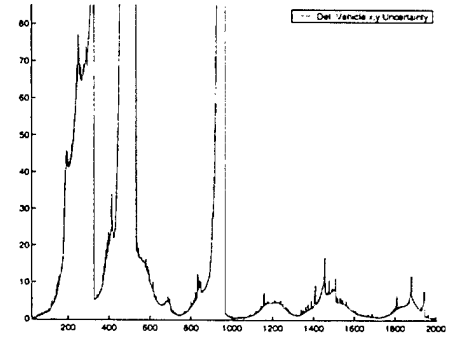
Determinant of Vehicle Uncertainty



Close-up of Determinant



Determinant of Vehicle x,y Uncertainty



Close-up of Determinant

Figure 4-10: Concurrent Mapping and Localization. **Top Left:** The determinant of the vehicle uncertainty, $\mathbf{P}_{vv}$. This determinant converges to the determinant of the vehicle uncertainty at the time the first feature was observed. **Top Right:** The close-up shows the convergence of the uncertainty. **Bottom Left:** The determinant of x and y vehicle uncertainty, $\mathbf{P}_{xy}$. **Bottom Right:** The close-up shows that in the limit, as successive observations are taken, the vehicle uncertainty converges to the initial x and y uncertainty of the vehicle at the time it first observed a feature.

4.2.3 Planned perception simulation #1

This simulation integrates planned perception and CML. The vehicle navigates from waypoint to waypoint. The planned perception algorithm goes into effect when the uncertainty in the vehicle estimate reaches a certain threshold. This simulation is broken down into scenarios demonstrating different strategies of integrating planned perception and CML.

As discussed in Section 3.4, planned perception chooses which feature the vehicle should steer toward defined by the equation

$$\Gamma_i = -\alpha \mid (\mathbf{W}_i \mathbf{S}_i \mathbf{W}_i^T)_v \mid + \beta \|\hat{\mathbf{x}}_{fi} - \hat{\mathbf{x}}_v\| + \zeta \cdot f(density_{fi}). \tag{4.5}$$

The action of the robot is given by evaluating Equation 4.6 using the metric Equation 4.7. The vehicle steers towards the feature which satisfy

$$\mathbf{\Lambda}_{i*} = arg_i \; min \; \mathbf{\Gamma}_i, \tag{4.6}$$

$$\mathbf{\Lambda}_{i*}(k) \leq \mathbf{\Lambda}_{i*}(k-1). \tag{4.7}$$

The first scenario addresses only the first criteria for planned perception: For all mapped features $i = 1, \ldots, N_f$, the re-observation of which feature i will best improve vehicle estimates? The second scenario addresses: What is the cost of re-observing feature i ? The third scenario combines both of these criteria along with addressing the data association problem.

For each scenario, the uncertainty determinant threshold is set at $\mathbf{P}_{vvthreshold} = 0.05$.

Planned perception - Scenario 1

This scenario addresses the first criteria of planned perception. It only address $\mid (\mathbf{W}_i \mathbf{S}_i \mathbf{W}_i^T)_v \mid$ Equation 4.5. Thus, the weighting gains β and ζ are set to zero. The cost and risk of associated with re-observing feature i are not addressed in this

scenario.

The trajectories of the true and estimated vehicle paths are shown in Figure 4-11. The vehicle deviates from the desired path, Figure 4-2, when the uncertainty for the vehicle exceeds the given threshold. When this occurs, the vehicle chooses to steer toward feature i that satisfies

$$\Gamma_i = -\alpha | \ (\mathbf{W}_i \mathbf{S}_i \mathbf{W}_i^T)_v \ |, \tag{4.8}$$

where α is set to 10^{21} to accurately compare the determinants of $(\mathbf{W}_i \mathbf{S}_i \mathbf{W}_i^T)_v$. This gain was set after analyzing the determinant values associated with Equation 4.5.

By deviating from the desired path, the vehicle is able to re-observe features that would otherwise not have been in the sensor's field of view. This allows for the vehicle to decrease its uncertainty in estimated states through the re-observation of features. The re-observation of features with a relatively small position uncertainty clearly provides more benefit than observing features with relatively larger uncertainties in position.

Figure 4-14 represents when the uncertainty threshold is breached. The threshold is plotted with the current uncertainty. The feature the vehicle steers toward is reflected in Figure 4-15. The vehicle tends to steer toward features that are mapped earlier during the mission. This is expected because these features have less uncertainty in their position estimates and provide better localization points for vehicle state estimation.

These results are consistent with the convergence theorems described in Section 2.5. The first convergence theorem states: The determinant of any sub-matrix of the map covariance matrix decreases monotonically as successive observations are made. This means that the error in the estimates of the absolute location of the vehicle and features diminishes as successive measurements are made. This is the point of planned perception; when the vehicle uncertainty reaches a certain threshold, the vehicle chooses to re-observe certain features in order to reduce the uncertainty in its estimates.

In this scenario, the reason the vehicle decides to head toward the features with less uncertainty associated with their position is directly related to the second and third convergence theorems described in Section 2.5. As successive observations are made, the uncertainty in feature location diminishes to the uncertainty that was present in the vehicle at the time that feature was first observed. Thus, the vehicle chooses to steer toward features having relatively small uncertainties.

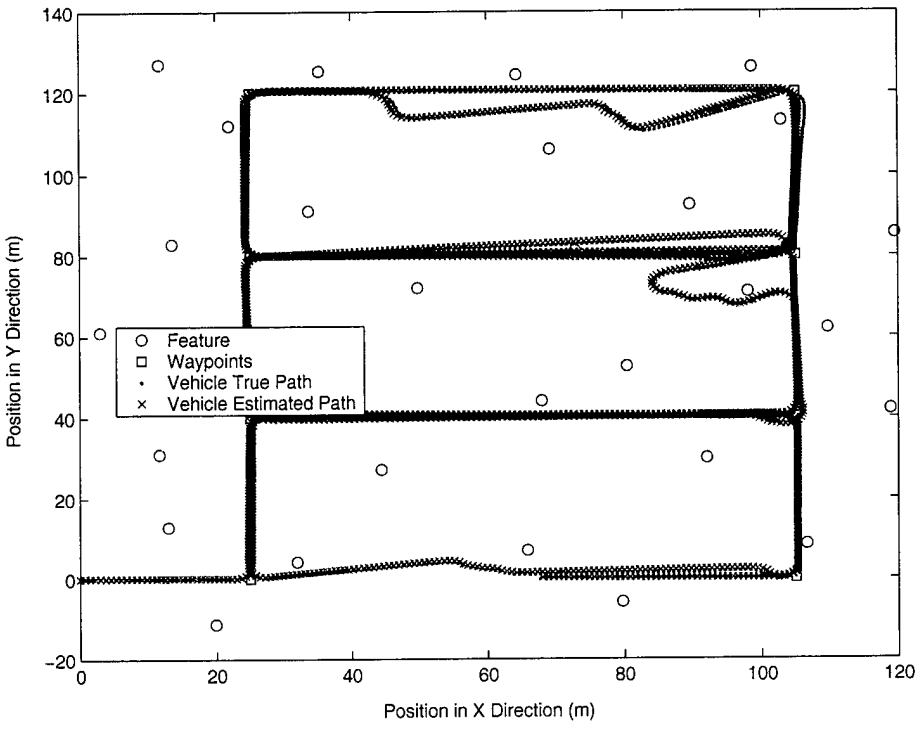


Figure 4-11: Planned Perception, Scenario 1. True vehicle path versus estimated vehicle path. The red crosses represent the true vehicle path. The blue dots follow the estimated vehicle path. The minor jumps in estimated and true position occur when the vehicle obtains an update and improves its estimated position.

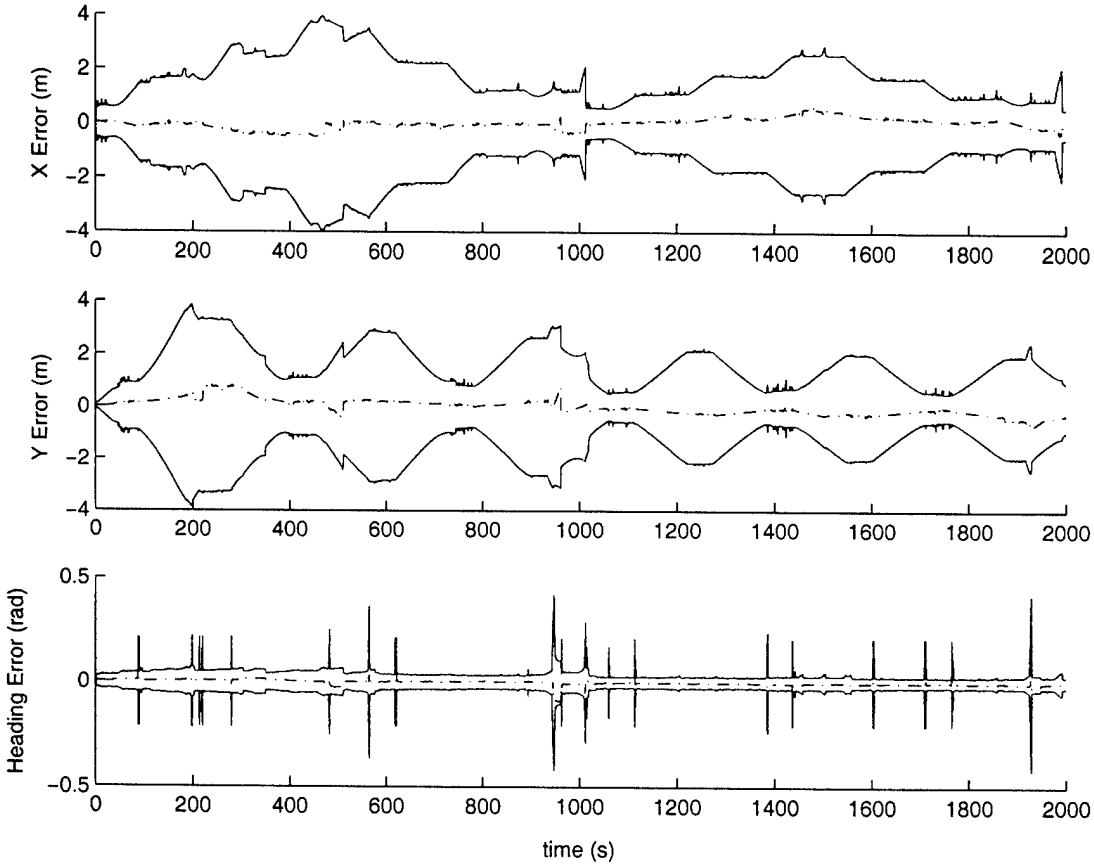


Figure 4-12: Planned Perception, Scenario 1. The errors of the state estimate when performing planned perception. The actual errors are represented by the red dashed line. The solid blue lines represent $\pm 1\sigma$ error bounds.

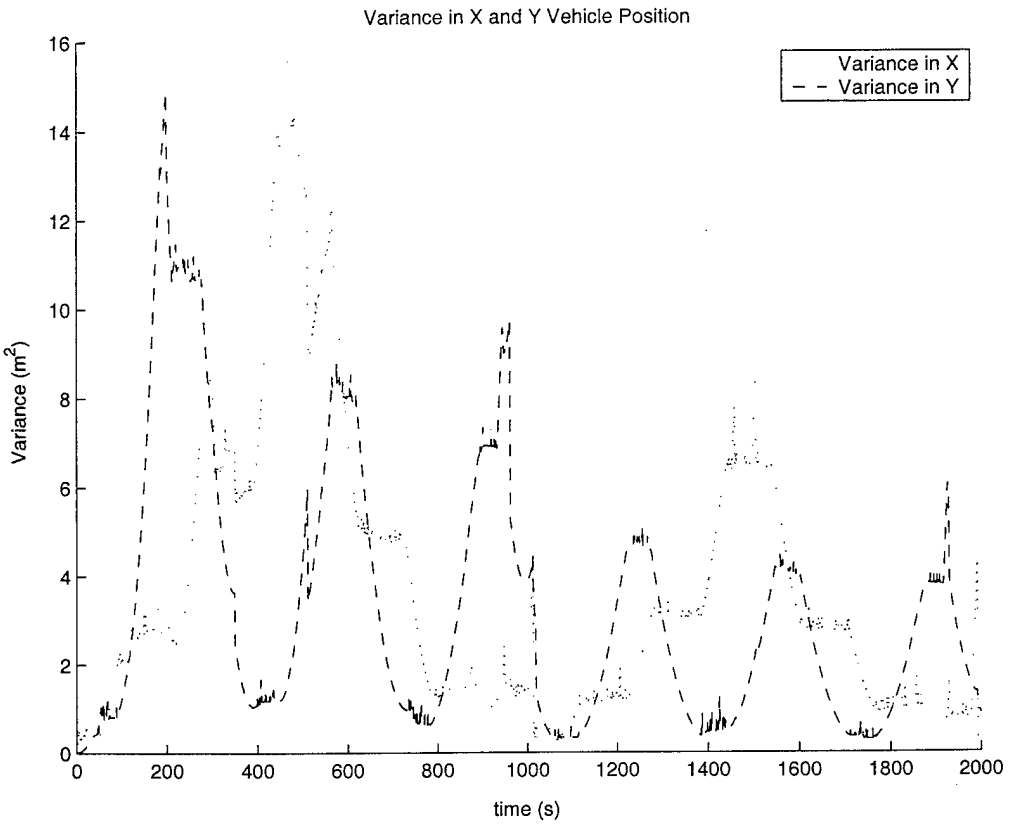
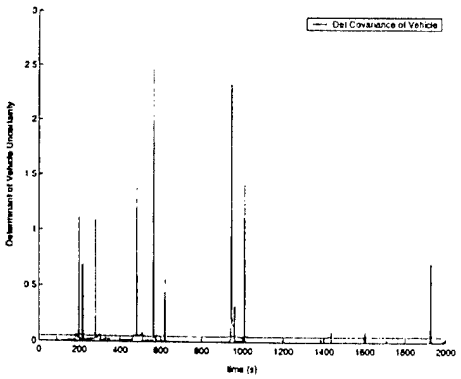
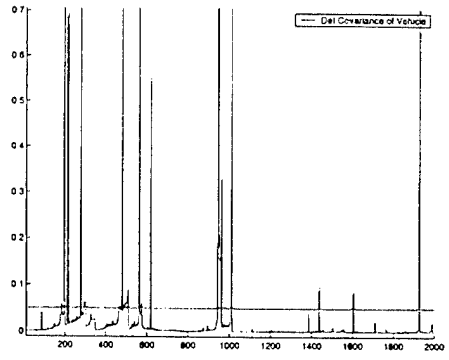


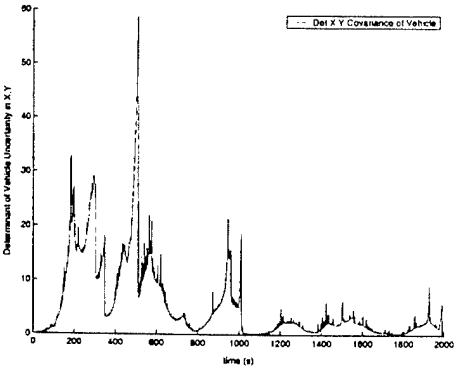
Figure 4-13: Planned Perception, Scenario 1. The variances in x and y position are plotted. The dotted red line is the variance in x while the dashed blue line is the variance in y. As expected, these variances converge to lower limits.



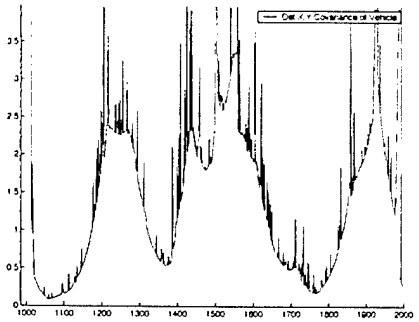
Determinant of Vehicle Uncertainty



Close-up of Determinant

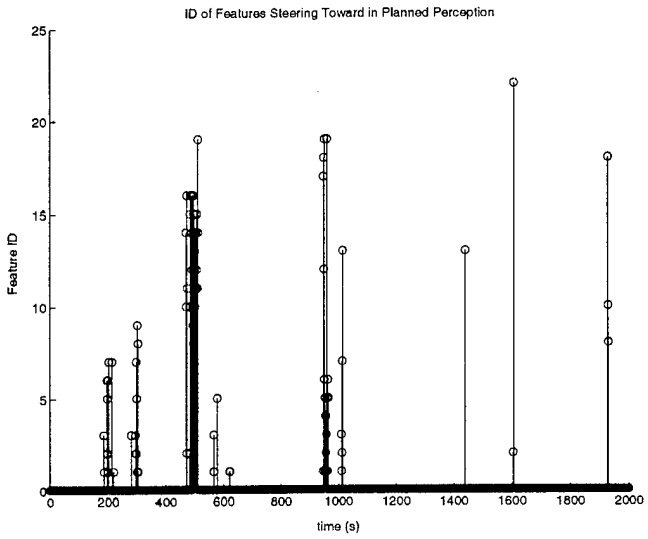


Determinant of Vehicle x,y Uncertainty

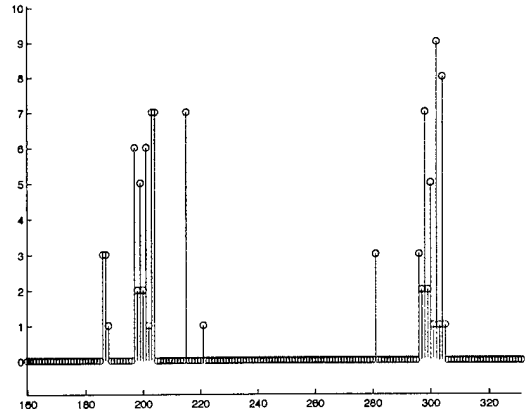


Close-up of Determinant

Figure 4-14: Planned Perception, Scenario 1. **Top left:** Determinant of vehicle uncertainty, $\mathbf{P}_{vv}$. The black line represents the threshold when planned perception was performed. **Top right:** The uncertainty is quickly restored below the threshold as a result of the planned perception algorithm. The close-up also shows that in the limit, as successive observations are taken, the vehicle uncertainty converges to the initial uncertainty of the vehicle at the time it first observed a feature. This is also below the threshold. **Bottom:** The same is shown for the x and y uncertainty of the vehicle, $\mathbf{P}_{xy}$.



IDs of Features Vehicle Chose to Steer Toward



Close-up of Feature IDs

Figure 4-15: Planned Perception, Scenario 1. **Top:** This figure shows the IDs of the features the vehicle chose to steer toward while performing planned perception. The IDs are obtained as the feature is mapped (i.e. the first feature mapped gets ID #1). The feature ID is 0 when the vehicle is "exploring" and the uncertainty is below the given threshold. **Bottom:** A close-up of the feature ID figure is shown here.

Planned perception - Scenario 2

This scenario addresses the second criteria of planned perception. Referring to Equation 4.5, this scenario only address $\|\hat{\mathbf{x}}_{fi} - \hat{\mathbf{x}}_v\|$. Thus, the weighting gains α and ζ are set to zero.

In this scenario, planned perception is performed by minimizing the distance $\|\hat{\mathbf{x}}_{fi} - \hat{\mathbf{x}}_v\|$. Therefore, while performing planned perception, the vehicle steers toward the closest, previously mapped feature. Thus, when the threshold uncertainty determinant is exceeded, the vehicle steers toward feature i that satisfies

$$\Gamma_i = \beta\|\hat{\mathbf{x}}_{fi} - \hat{\mathbf{x}}_v\|, \tag{4.9}$$

where β is set to 1 to accurately compare the distance $\|\hat{\mathbf{x}}_{fi} - \hat{\mathbf{x}}_v\|$.

Deviating from the desired path allows the vehicle to re-observe features that otherwise would not have been in the sensor's field of view. In this case, the vehicle steers toward the closest feature previously mapped.

Because the constraint is to steer toward the feature closest in distance, the vehicle does not always diminish its uncertainty through the re-observation of feature i that satisfies Equation 4.9. Heading toward and re-observing the closest feature sometimes causes the vehicle to "circle" a feature, re-observing it time and time again. Thus, the uncertainty does not always achieve the desired reduction through the re-observation of feature i. The "circling" of features can be shown in Figure 4-16.

The vehicle is able to reduce its uncertainty below the given threshold, not always through the re-observation of the closest feature, but through re-observing another feature while it is "circling." This is shown in Figure 4-20 and Figure 4-16 where the vehicle is constantly steering toward and circling a certain feature. The vehicle happens to re-observe another feature in the process of circling the desired feature i. The re-observation of another feature, along with re-observing the feature closest to the vehicle, provide the update needed to drive vehicle uncertainty below the threshold.

Figure 4-19 represents the breaching of the uncertainty threshold. The threshold

is plotted with the current uncertainty. Figure 4-18 shows the variances of x and y vehicle position. As successive re-observations of featuresa are made, these variances converge as expected.

The criteria set forth in this scenario does not always provide the necessary means to achieve the desired vehicle uncertainty level. This is due to the vehicle steering toward the mapped feature that is closest in distance. It is not always the re-observation of the closest feature i that provides the necessary update information; it is re-observing feature i along with re-observing a different feature in the process of successive acts of circling, that provide the necessary measurements to successfully reduce vehicle uncertainty.

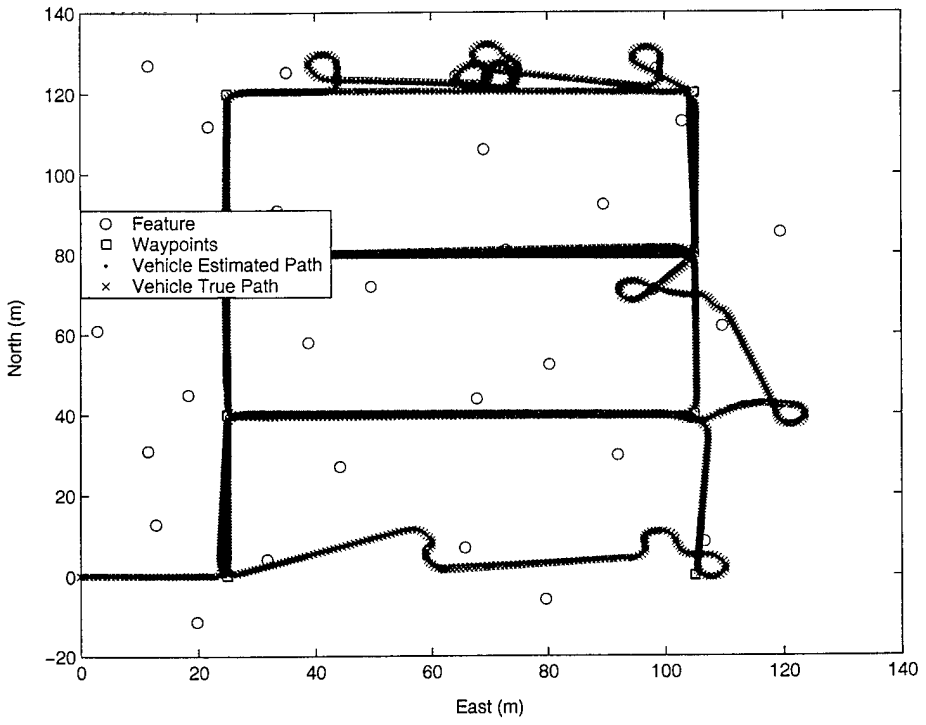


Figure 4-16: Planned Perception, Scenario 2. True vehicle path versus estimated vehicle path. The red crosses represent the true vehicle path. The blue dots follow the estimated vehicle path. The loops in the path show where the vehicle "circles" a feature.

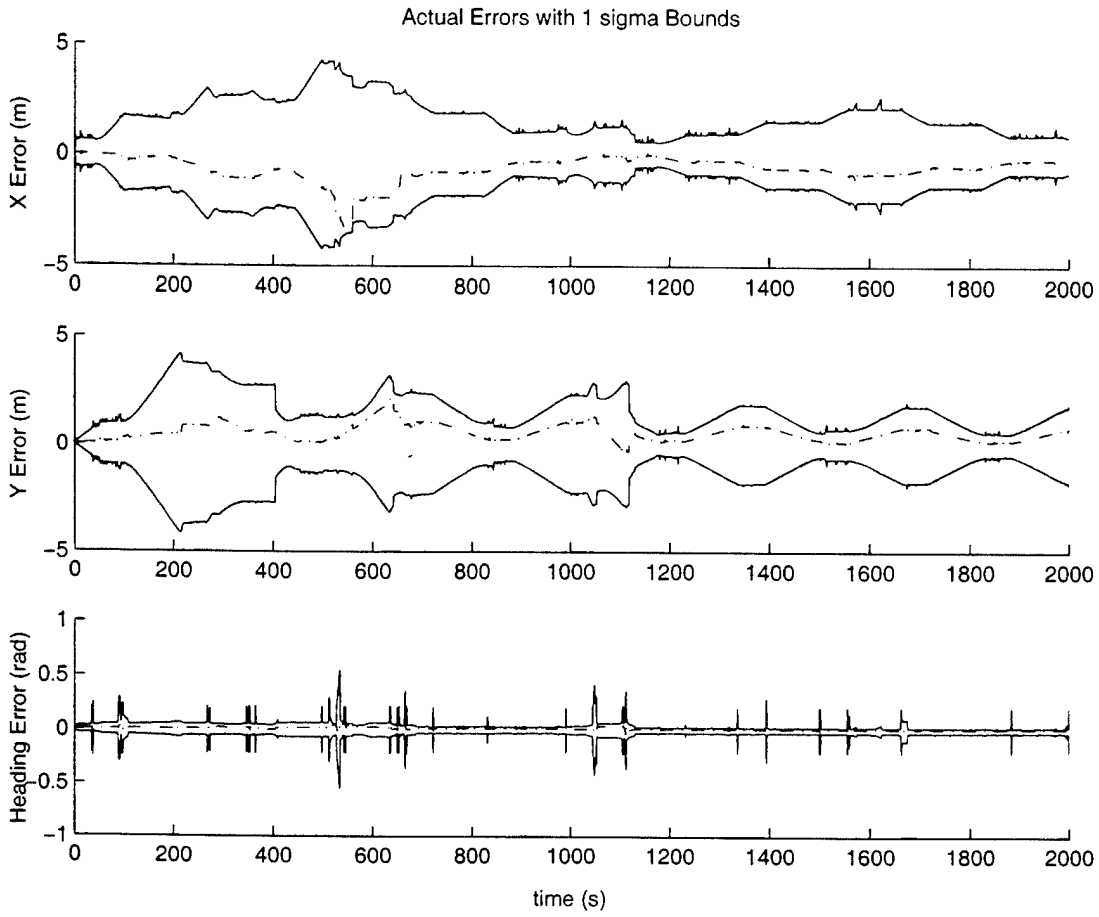


Figure 4-17: Planned Perception, Scenario 2. The errors of the state estimate when performing planned perception. The actual errors are represented by the red dashed line. The solid blue lines represent $\pm 1\sigma$ error bounds.

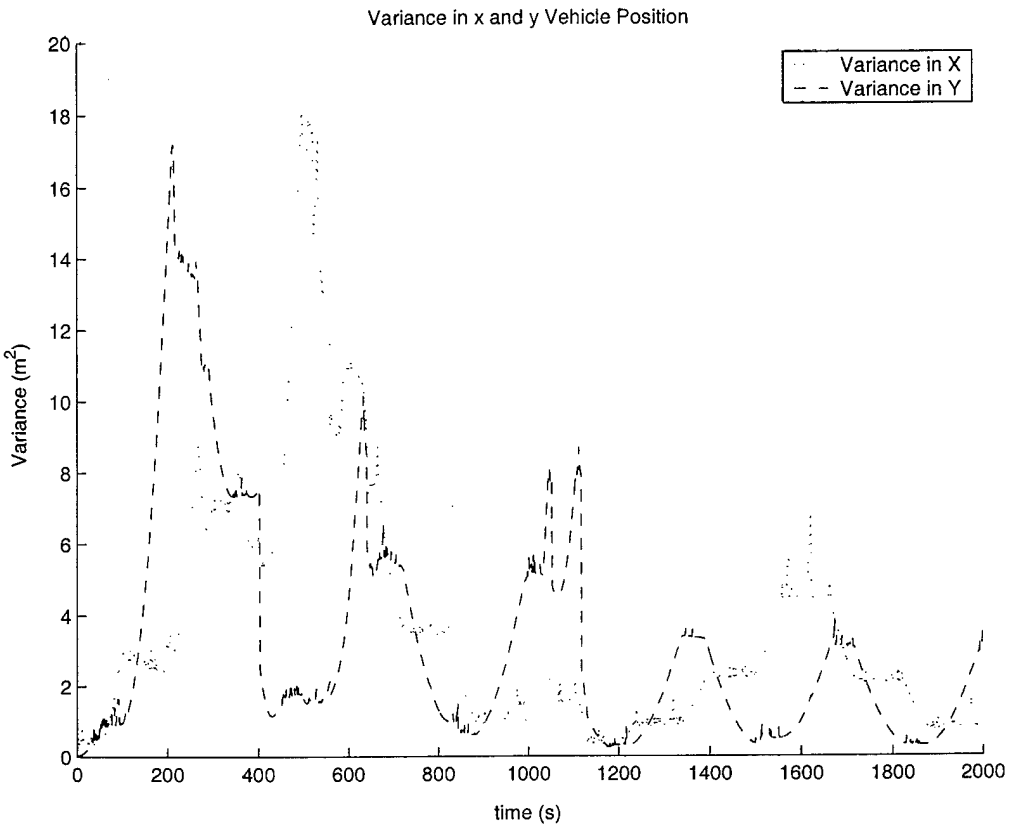
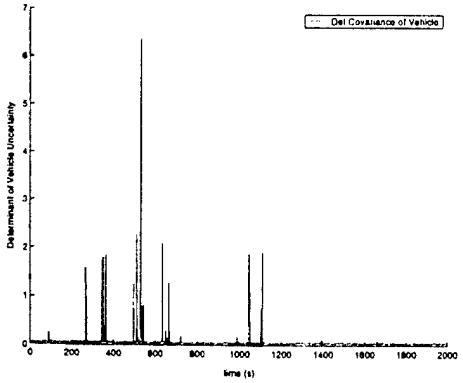
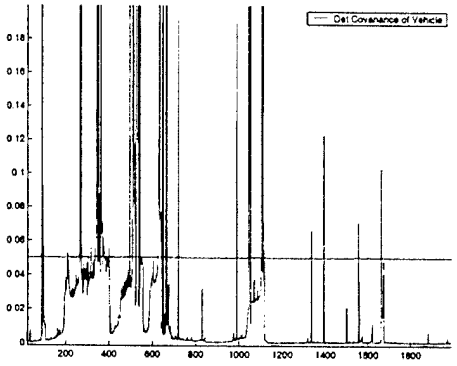


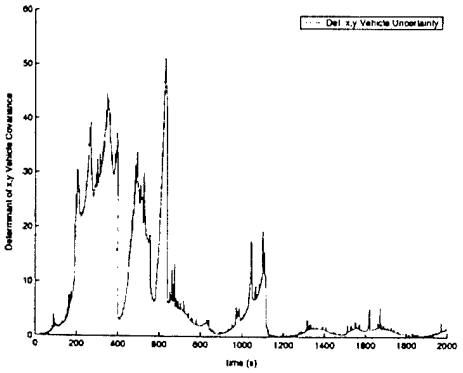
Figure 4-18: Planned Perception, Scenario 2. The variances in x and y position are plotted. The red dotted line is the variance in x while the dashed blue line is the variance in y. As expected, these variances converge to lower limits.



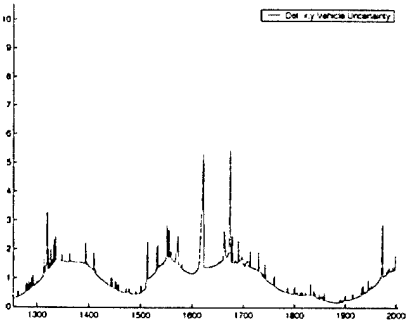
Determinant of Vehicle Uncertainty



Close-up of Determinant

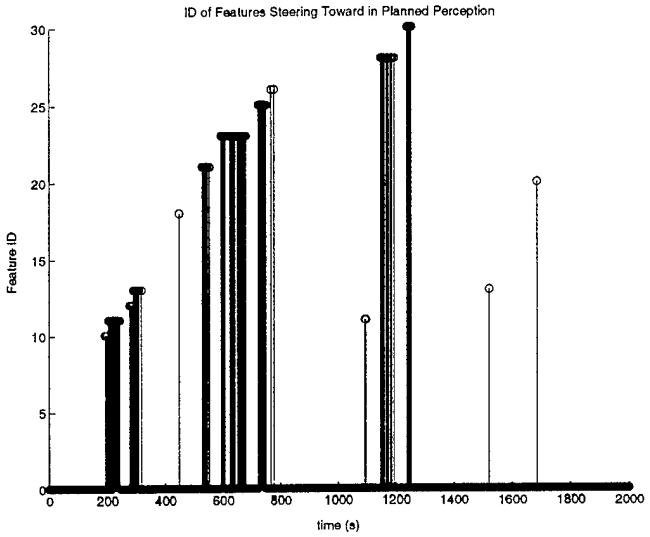


Determinant of Vehicle x,y Uncertainty

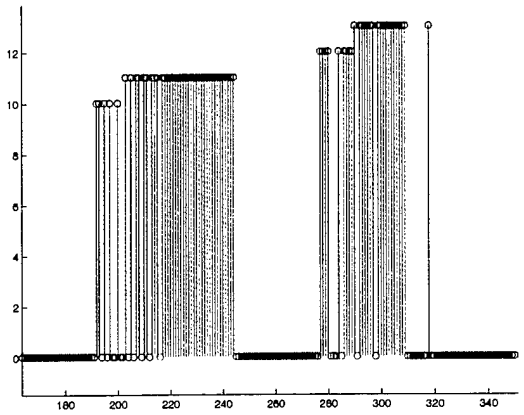


Close-up of Determinant

Figure 4-19: Planned Perception, Scenario 2. **Top left:** Determinant of vehicle uncertainty, $\mathbf{P}_{vv}$. The black line represents the threshold when planned perception was performed. **Top right:** The overall uncertainty is quickly restored below the threshold as a result of the planned perception algorithm. The close-up also shows that in the limit, as successive observations are taken, the vehicle uncertainty converges to the initial uncertainty of the vehicle at the time it first observed a feature. This is also below the threshold. **Bottom:** The same is shown for the x and y uncertainty of the vehicle, $\mathbf{P}_{xy}$.



IDs of Features Vehicle Chose to Steer Toward



Close-up of Feature IDs

Figure 4-20: **Top:** Planned Perception, Scenario 2. This figure shows the IDs of the features the vehicle chose to steer toward while performing planned perception. The IDs are obtained as the feature is mapped (i.e. the first feature mapped gets ID #1). The feature ID is 0 when the vehicle is "exploring" and the uncertainty is below the given threshold. **Bottom:** A close-up of the feature ID figure is shown here. The close-up shows how the vehicle continually circles a feature until the uncertainty decreases below the given threshold.

Planned perception - Scenario 3

This scenario combines all three criteria of planned perception. Thus, the feature the vehicle chooses to steer toward while performing planned perception is determined by evaluating

$$\Gamma_i = -\alpha|\ (\mathbf{W}_i\mathbf{S}_i\mathbf{W}_i^T)_v\ | + \beta\|\hat{\mathbf{x}}_{fi} - \hat{\mathbf{x}}_v\| + \zeta \cdot f(density_{fi}), \tag{4.10}$$

$$\mathbf{\Lambda}_{i*} = arg_i\ min\ \mathbf{\Gamma}_i, \tag{4.11}$$

based on the metric

$$\mathbf{\Lambda}_{i*}(k) \leq \mathbf{\Lambda}_{i*}(k-1). \tag{4.12}$$

Since all three criteria discussed in Chapter 3 are being considered simultaneously, Γ_i must be scaled by the weighting factors to obtain a scalar number that accurately weights all three factors according to the desired mission. For this scenario, the weighting factor ζ is scaled to 20. Thus, the factor weights Equation 4.10 so that the vehicle will be more likely to steer away from any feature that has another feature within 10 meters of it. This criteria is set to minimize the problems inherent in data association. The weighting gains α and β are set to 10^{21} and 2, respectively. These gains affect the weighting of the criteria:

1. For all mapped features $i = 1, \ldots, N_f$, which re-observation of feature i will best improve vehicle estimates?

2. What is the cost of re-observing feature i ?

The values of the gains were chosen so as to weight each criteria equally. The values were obtained through analyzing values obtained by evaluating Equation 4.10 in simulation.

Figure 4-21 shows the path the vehicle travelled. The vehicle deviates from the desired path in order to re-observe features and reduce the estimated uncertainty.

The path travelled reflects the criteria for this scenario; the vehicle steers toward a combination of: features with less certainty, features that are relatively close to the current vehicle position, and features not located in cluttered areas. This is reflected in Figure 4-25 as the vehicle steers toward a mixture of features.

By performing planned perception, the uncertainty of the vehicle is desired to be held below a given threshold. Figure 4-24 shows the vehicle uncertainty and the given threshold. This is also reflected in the errors associated with each state estimate, Figure 4-22, and the variance of the vehicle's x and y estimates, Figure 4-23. As the vehicle uncertainty exceeds the threshold, the vehicle maneuvers to re-observe features.

Through the combination of all three criteria, the vehicle attempts to maintain a desired uncertainty level by re-observing features. The action of the robot is given by evaluating Equation 4.11 using the metric Equation 4.12. The vehicle steers towards the feature which satisfies the above criteria. The features are chosen based on a tradeoff between all three criteria. The features chosen are a combination of features that are relatively close to the vehicle, features not located in a cluttered area, and features having relatively small uncertainties.

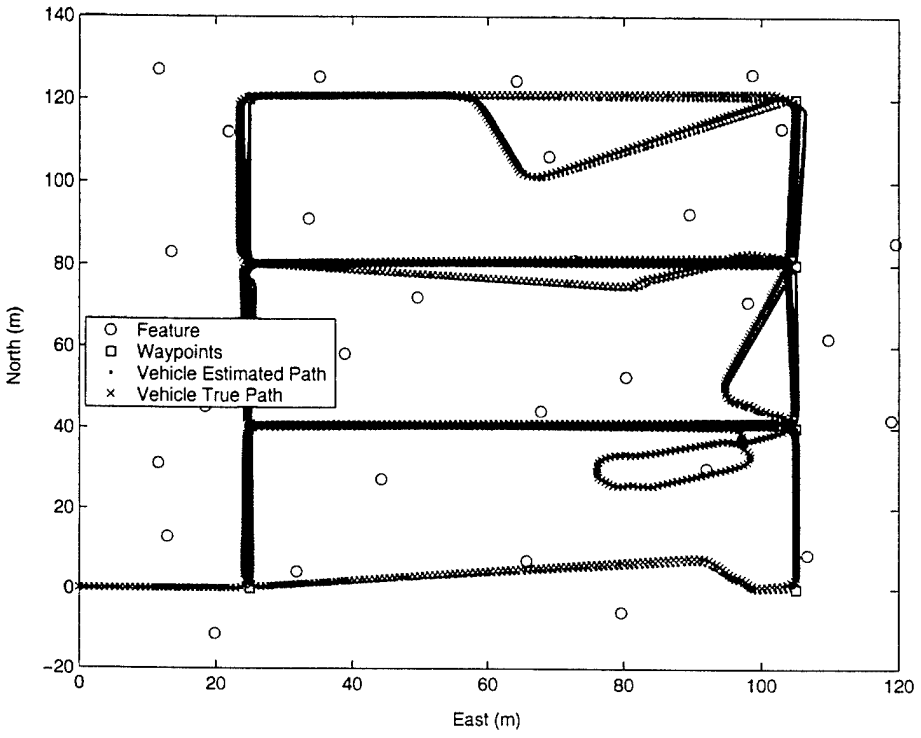


Figure 4-21: Planned Perception, Scenario 3. True vehicle path versus estimated vehicle path. The red crosses represent the true vehicle path. The blue dots follow the estimated vehicle path.

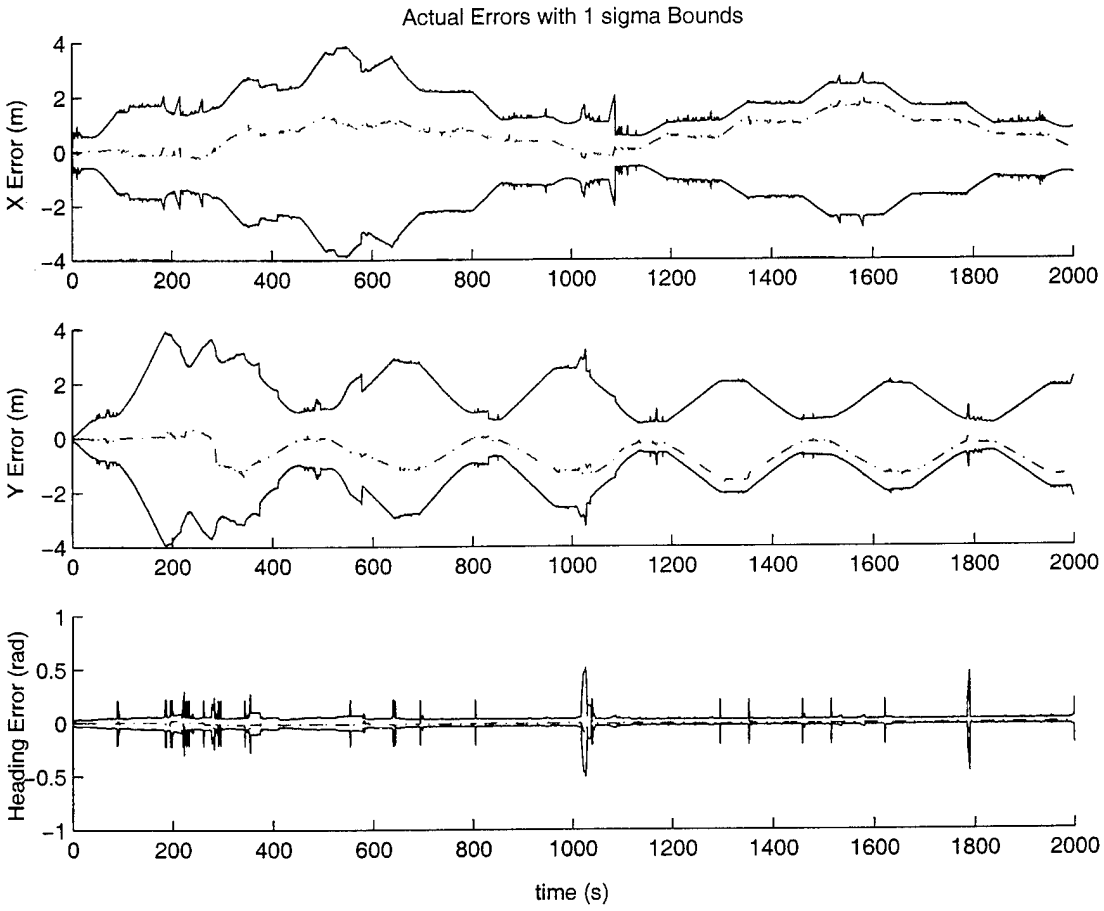


Figure 4-22: Planned Perception, Scenario 3. The errors of the state estimate when performing planned perception. The actual errors are represented by the red dashed line. The solid blue lines represent $\pm 1\sigma$ error bounds.

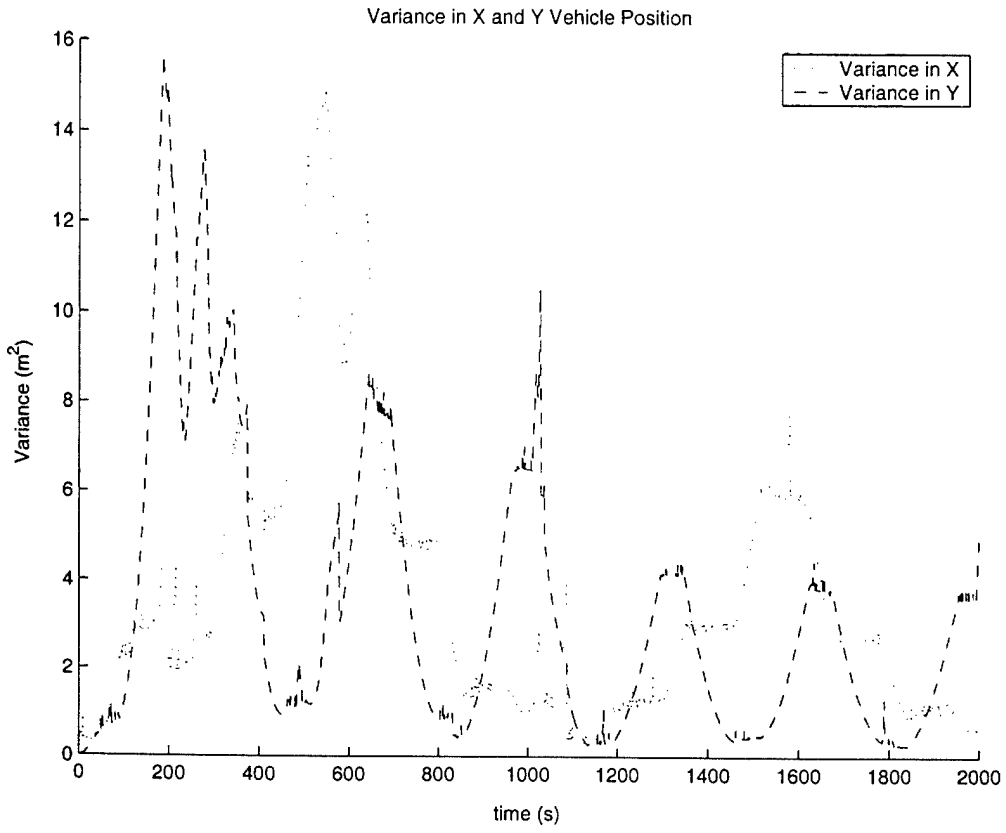
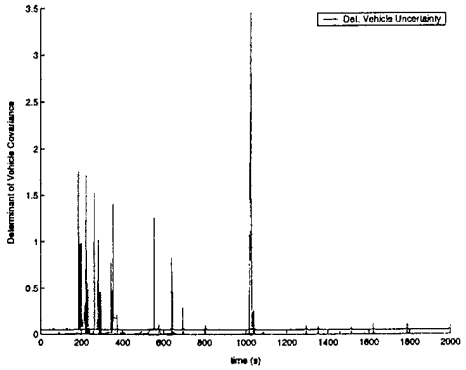
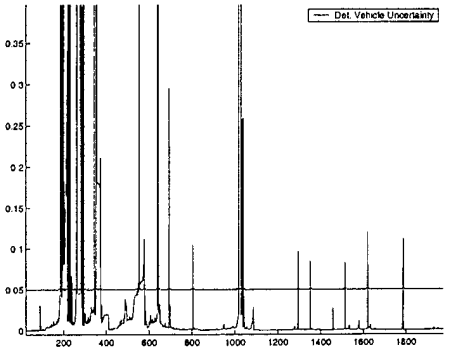


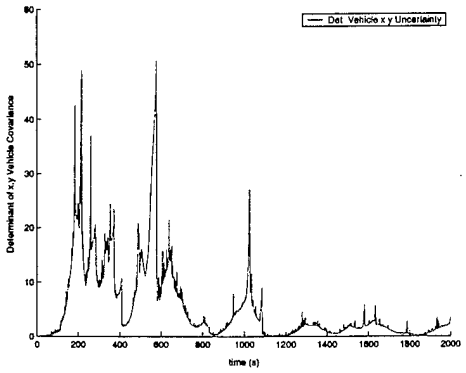
Figure 4-23: Planned Perception, Scenario 3. The variances in x and y position are plotted. The red dotted line is the variance in x while the dashed blue line is the variance in y. As expected, these variances converge to lower limits.



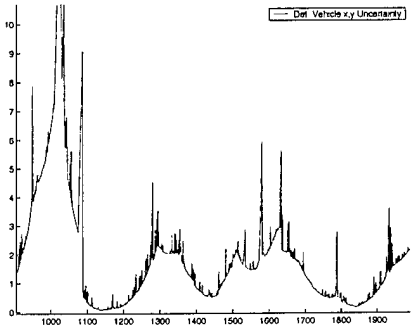
Determinant of Vehicle Uncertainty



Close-up of Determinant

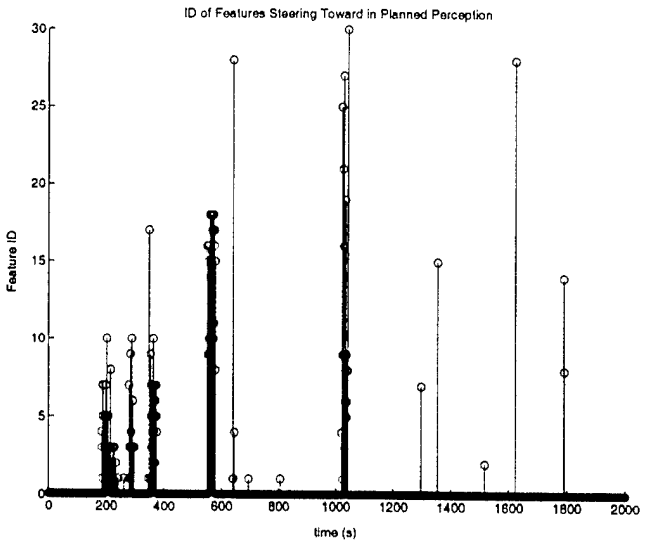


Determinant of Vehicle x,y Uncertainty

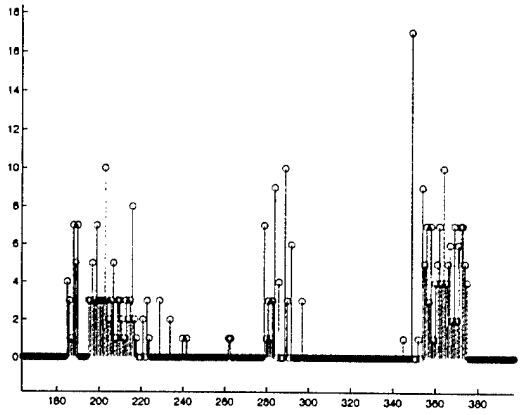


Close-up of Determinant

Figure 4-24: Planned Perception, Scenario 3. **Top left:** Determinant of vehicle uncertainty, $\mathbf{P}_{vv}$. The black line represents the threshold when planned perception was performed. **Top right:** The close-up shows that in the limit, as successive observations are taken, the vehicle uncertainty converges to the initial uncertainty of the vehicle at the time it first observed a feature. The vehicle uncertainty is quickly restored below the threshold as a result of the planned perception algorithm. **Bottom left:** Determinant of vehicle x and y uncertainty, $\mathbf{P}_{xy}$. **Bottom Right:** As expected, the uncertainty converges.



IDs of Features Vehicle Chose to Steer Toward



Close-up of Feature IDs

Figure 4-25: Planned Perception, Scenario 3. **Top:** This figure shows the IDs of the features the vehicle chose to steer toward while performing planned perception. The IDs are obtained as the feature is mapped (i.e. the first feature mapped gets ID #1). The feature ID is 0 when the vehicle is "exploring" and the uncertainty is below the given threshold. **Bottom:** A close-up of the feature ID figure is shown here. The vehicle chooses to steer toward a combination of features that have relatively little uncertainty (those mapped earlier) and features that are close to its current position.

4.2.4 Planned perception simulation #2

This simulation integrate the planned perception and CML. Planned perception simulation #1, described in Section 4.2.3, constrains the overall uncertainty associated with the vehicle states, $\mathbf{P}_{vv}$. This simulation, however, constrains only the uncertainty associated with the vehicle's x and y estimates, termed $\mathbf{P}_{xy}$. This is chosen because it specifically constrains the area of uncertainty in the vehicle's x and y position estimates.

The vehicle desires to navigate from waypoint to waypoint. The planned perception algorithm goes into effect when the determinant of $\mathbf{P}_{xy}$ exceeds a given threshold $\mathbf{P}_{xythreshold}$. This threshold determinant is set to 4.0. The number 4.0 is chosen to reflect the desire to maintain the vehicle's x and y uncertainty below 4 m^2. This simulation is broken down into different scenarios. The following scenarios demonstrate different strategies of integrating planned perception and CML when constraining the error uncertainty for the x and y state estimates. Except for the determinant threshold constraining $\mathbf{P}_{xy}$ and being set to 4.0, the scenarios are the same as those in Section 4.2.3.

Planned perception - Scenario 1

This scenario addresses the first criteria of planned perception. Thus, this scenario only addresses $\mid (\mathbf{W}_i\mathbf{S}_i\mathbf{W}_i^T)_v \mid$ in Equation 4.5. The weighting gains β and ζ are set to zero and α is set to 10^{21} as in Section 4.2.3.

Figure 4-26 shows the trajectories of the true and estimated vehicle paths. When the vehicle's x and y uncertainty exceed the given threshold, the vehicle deviates from the desired path to re-observe features. The re-observation of features allows the vehicle to decrease its uncertainty in the x and y estimated states. In turn, this also decreases the overall vehicle uncertainty, $\mathbf{P}_{vv}$. This can be seen in Figure 4-29.

While performing planned perception, the vehicle steers towards features as shown in Figure 4-30. As expected, these features tend to be those mapped earlier during the mission. This is because these features have more certainty in their position estimates

and will provide better updates for vehicle state estimation.

The uncertainty threshold is breached mainly during the vehicle's first cycle of the "lawnmower" pattern. This is shown in Figure 4-26 and also in the close-up of Figure 4-30. As the close-up shows, the vehicle performs in localization mode mainly during the first run of the pattern (up to around 900 seconds). The vehicle then re-observes features reducing uncertainty estimates. After closing-the-loop, the vehicle mainly operates in exploration mode for the rest of the mission.

As described in Section 4.2.3 scenario 1, these results are consistent with the convergence theorems described in Section 2.5.

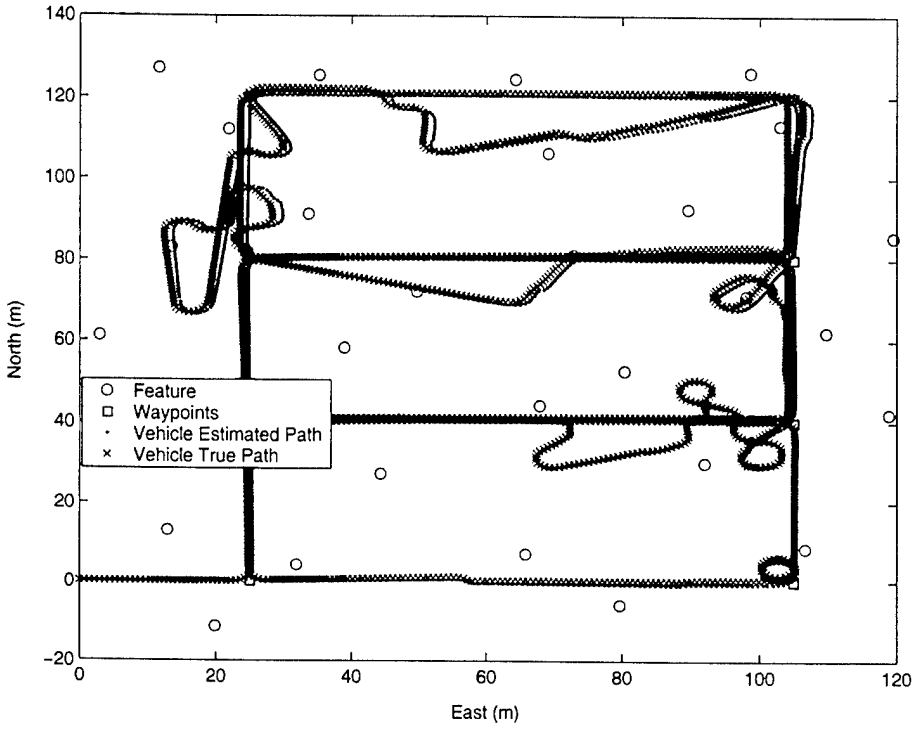


Figure 4-26: Planned Perception, Scenario 1. True vehicle path versus estimated vehicle path. The red crosses represent the true vehicle path. The blue dots follow the estimated vehicle path. The minor jumps in estimated and true position occur when the vehicle obtains an update and improves its estimated position.

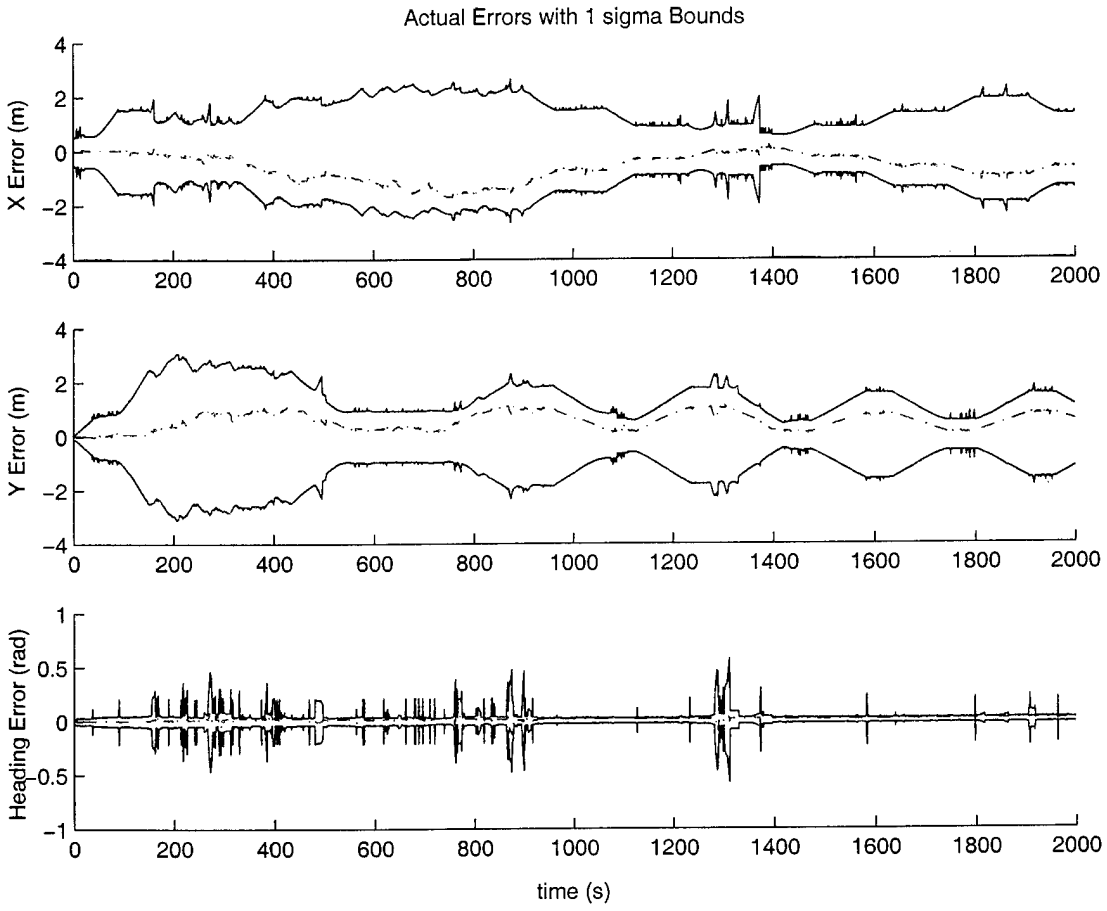


Figure 4-27: Planned Perception, Scenario 1. The errors of the state estimate when performing planned perception. The actual errors are represented by the red dashed line. The solid blue lines represent $\pm 1\sigma$ error bounds. As expected, these errors are bounded.

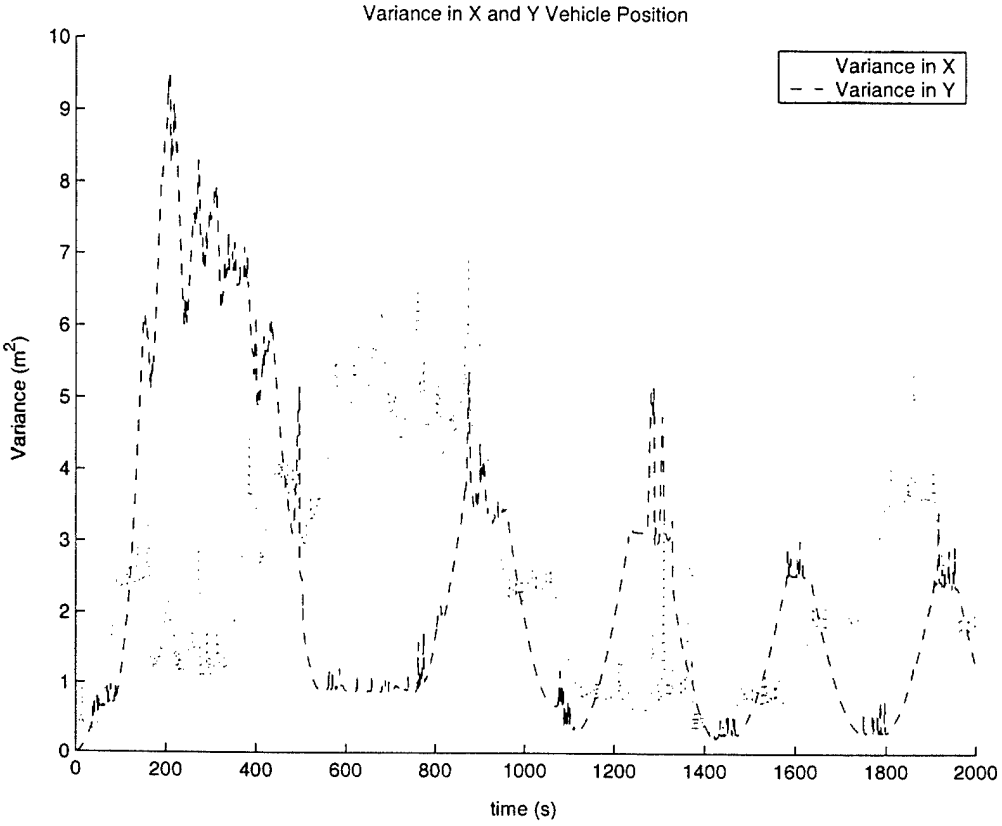
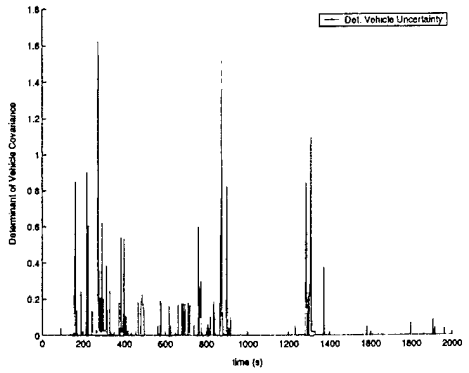
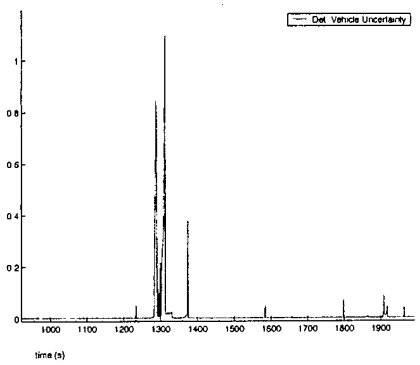


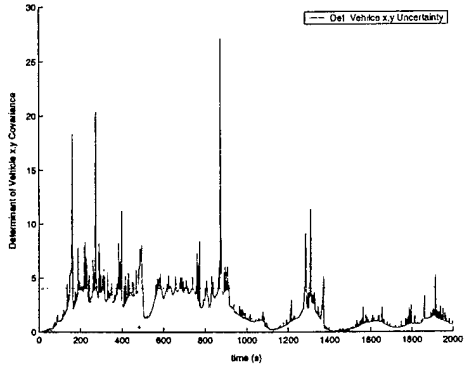
Figure 4-28: Planned Perception, Scenario 1. The variances in x and y position are plotted. The red dotted line is the variance in x while the dashed blue line is the variance in y. As expected, these variances converge to lower limits.



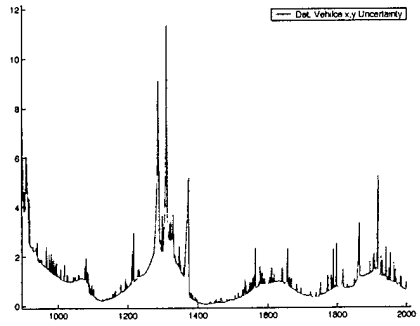
Determinant of Vehicle Uncertainty



Close-up of Determinant

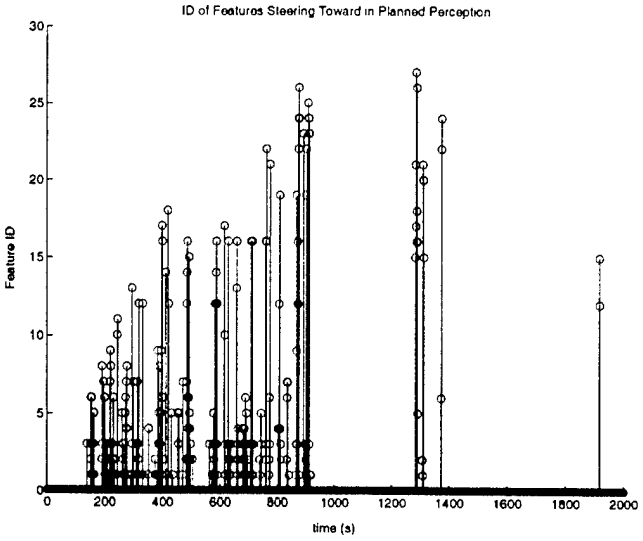


Determinant of Vehicle x,y Uncertainty

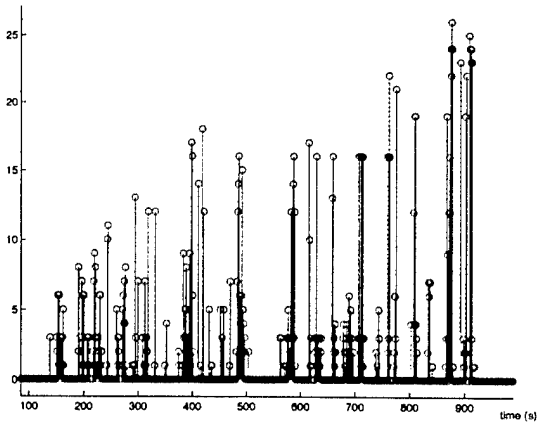


Close-up of Determinant

Figure 4-29: Planned Perception, Scenario 1. **Top left:** Determinant of vehicle uncertainty, $\mathbf{P}_{vv}$. **Top right:** The close-up shows the vehicle uncertainty converging to the initial vehicle uncertainty. **Bottom Left:** The determinant of the vehicle x and y uncertainty, $\mathbf{P}_{xy}$. The black line represents the threshold when planned perception was performed. **Bottom Right:** The uncertainty is quickly restored below the threshold as a result of the planned perception algorithm. The close-up also shows that in the limit, as successive observations are taken, the x and y vehicle uncertainty converges to the initial x,y uncertainty of the vehicle at the time it first observed a feature. This is also below the threshold.



IDs of Features Vehicle Chose to Steer Toward



Close-up of Feature IDs

Figure 4-30: Planned Perception, Scenario 1. **Top:** This figure shows the IDs of the features the vehicle chose to steer toward while performing planned perception. The IDs are obtained as the feature is mapped (i.e. the first feature mapped gets ID #1). The feature ID is 0 when the vehicle is "exploring" and the uncertainty is below the given threshold. **Bottom:** A close-up of the feature ID figure is shown here.

Planned perception - Scenario 2

In this scenario, planned perception is performed by minimizing the distance $\|\hat{\mathbf{x}}_{fi} - \hat{\mathbf{x}}_v\|$. Thus, when the threshold uncertainty determinant is exceeded, the vehicle steers toward feature i that satisfies Equation 4.5. The weighting gain β is set to 1 while the gains α and ζ are set to zero.

While performing planned perception, the vehicle heads toward the closest feature. The vehicle steers to re-observe this feature to decrease its estimated uncertainty. However, because of this constraint, heading toward and re-observing the closest feature sometimes causes the vehicle to "circle" a feature as described in Section 4.2.3. The circling of features can be shown in Figure 4-31.

The vehicle reduces its uncertainty not always through the re-observation of the closest feature, but through re-observing another feature while it is "circling." Figure 4-35 and Figure 4-31 show the vehicle constantly steering toward and circling a feature.

During its first "lap" of the desired path, the vehicle tends to operate mostly in localization mode (planned perception). However, after re-observing one of the first features mapped (closing-the-loop), the vehicle's x and y uncertainty diminishes below the threshold. Confident in its x and y estimates, the vehicle then spends most of the mission operating in exploration mode. This is seen through Figure 4-35; around time 1300 seconds, the vehicle *explores*.

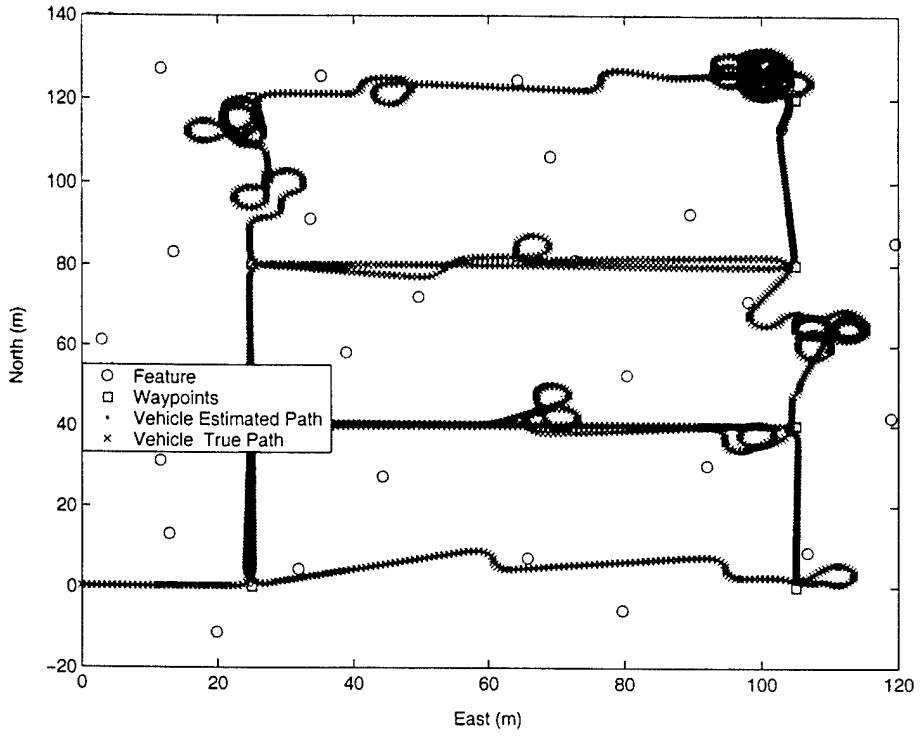


Figure 4-31: Planned Perception, Scenario 2. True vehicle path versus estimated vehicle path. The red crosses represent the true vehicle path. The blue dots follow the estimated vehicle path. The minor jumps in estimated and true position occur when the vehicle obtains an update and improves its estimated position. The loops in the path show where the vehicle "circles" a feature.

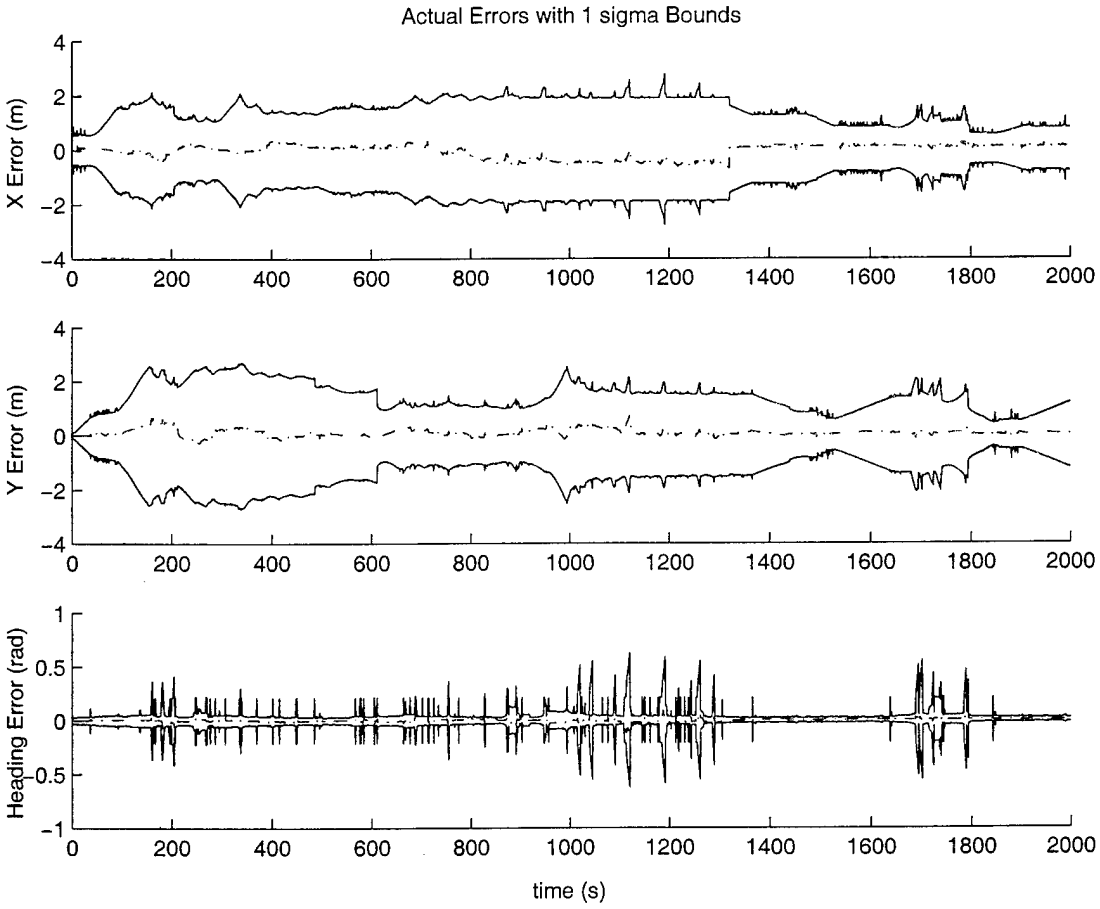


Figure 4-32: Planned Perception, Scenario 2. The errors of the state estimate when performing planned perception. The actual errors are represented by the red dashed line. The solid blue lines represent $\pm 1\sigma$ error bounds. As expected, these errors are bounded.

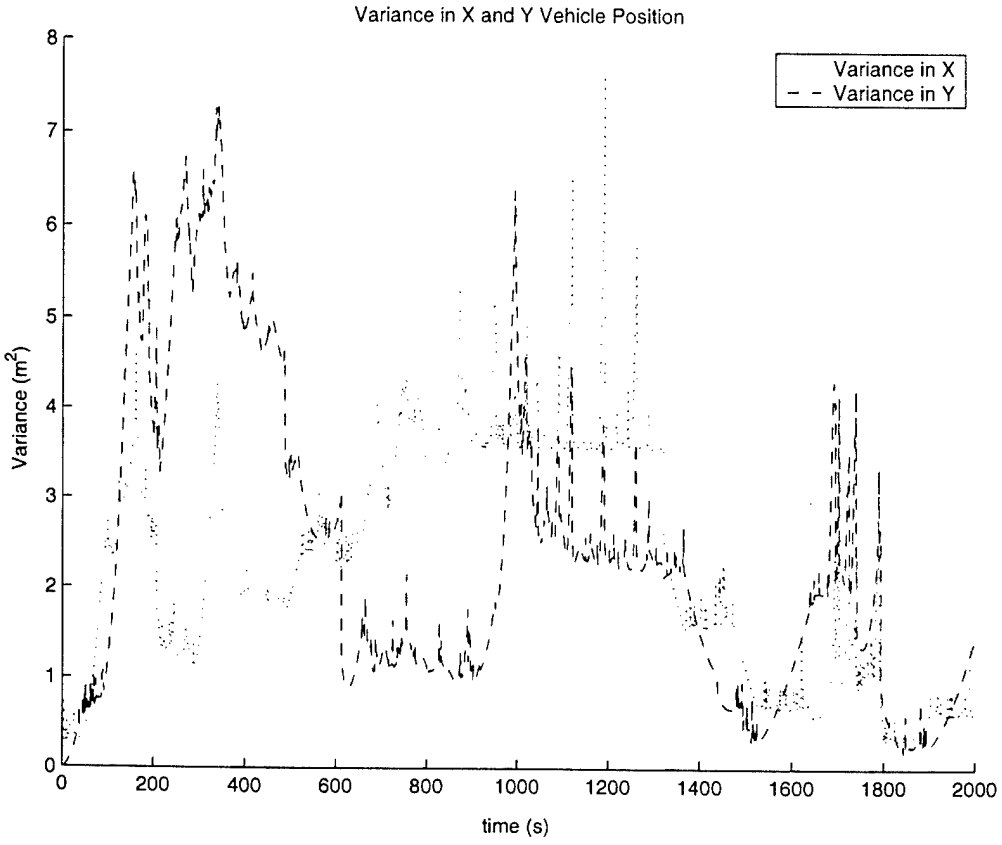
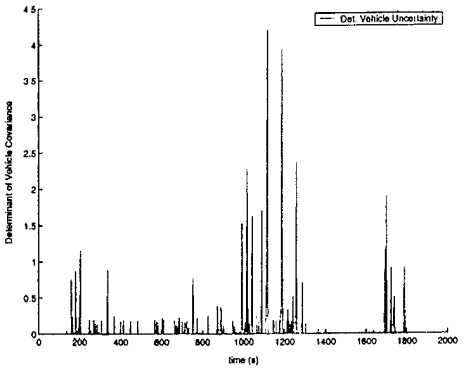
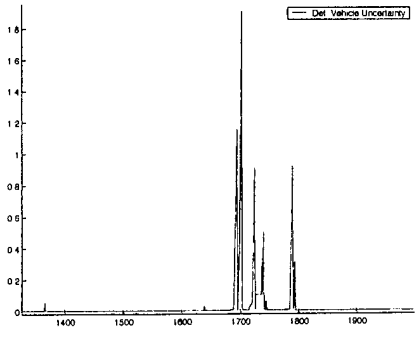


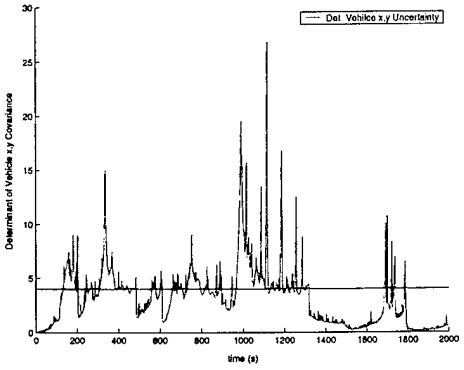
Figure 4-33: Planned Perception, Scenario 2. The variances in x and y position are plotted. The red dotted line is the variance in x while the dashed blue line is the variance in y. As expected, these variances converge to lower limits.



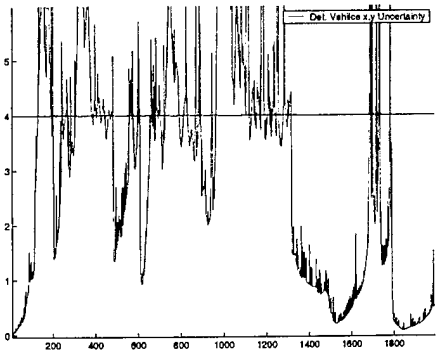
Determinant of Vehicle Uncertainty



Close-up of Determinant

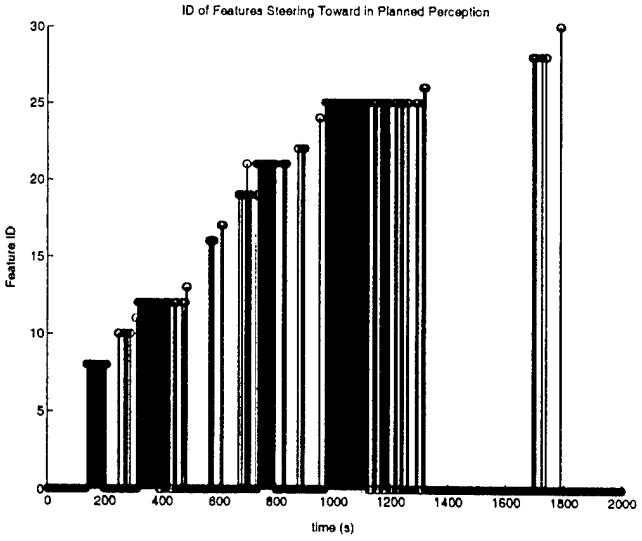


Determinant of Vehicle x,y Uncertainty

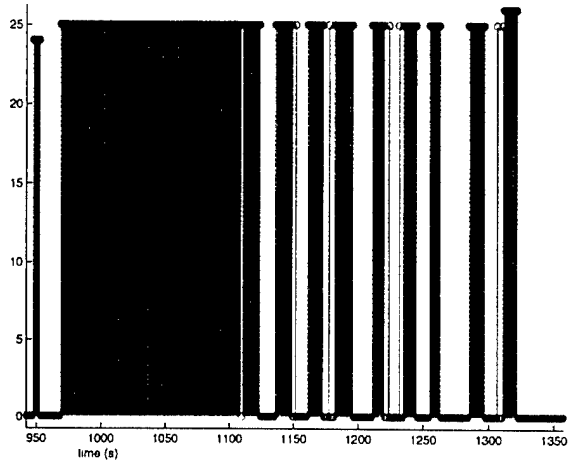


Close-up of Determinant

Figure 4-34: Planned Perception, Scenario 2. **Top left:** Determinant of vehicle uncertainty, $\mathbf{P}_{vv}$. **Top right:** The close-up shows the vehicle uncertainty converging to the initial vehicle uncertainty. **Bottom Left:** The determinant of the vehicle x and y uncertainty, $\mathbf{P}_{xy}$. The black line represents the threshold when planned perception was performed. **Bottom Right:** The uncertainty is quickly restored below the threshold as a result of the planned perception algorithm. The close-up also shows that in the limit, as successive observations are taken, the x and y vehicle uncertainty converges to the initial x,y uncertainty of the vehicle at the time it first observed a feature. This is also below the threshold.



IDs of Features Vehicle Chose to Steer Toward



Close-up of Feature IDs

Figure 4-35: Planned Perception, Scenario 2. **Top:** This figure shows the IDs of the features the vehicle chose to steer toward while performing planned perception. The IDs are obtained as the feature is mapped (i.e. the first feature mapped gets ID #1). The feature ID is 0 when the vehicle is "exploring" and the uncertainty is below the given threshold. **Bottom:** A close-up of the feature ID figure is shown here. The close-up shows how the vehicle continually circles a feature until the uncertainty decreases below the given threshold.

Planned Perception - Scenario 3

This scenario combines all three criteria of planned perception. The action of the robot is given by evaluating Equation 4.11. For this scenario, the weighting gains α, β, and ζ are set to 10^{21}, 2, and 20, respectively. The assignment of these values is discussed in Section 4.2.3.

The path the vehicle travelled is shown in Figure 4-36. The vehicle deviates from the desired path to re-observe features and reduce the uncertainty in the state estimates. The path travelled and features chosen to steer toward reflect the criteria for this scenario; while performing planned perception, the vehicle steers toward a feature that is a combination of: position uncertainty, distance to vehicle, and feature density. This is shown in Figure 4-40.

Planned perception is performed to maintain a desired vehicle uncertainty. In this simulation, the constraint is placed on the uncertainty in the vehicle's x and y position. Figure 4-39 shows the determinant of vehicle uncertainties with the given threshold. The desire to maintain a given uncertainty threshold is reflected in Figure 4-37, which shows the errors associated with each vehicle state estimate, and Figure 4-23, which shows the variance of the vehicle's x and y estimates.

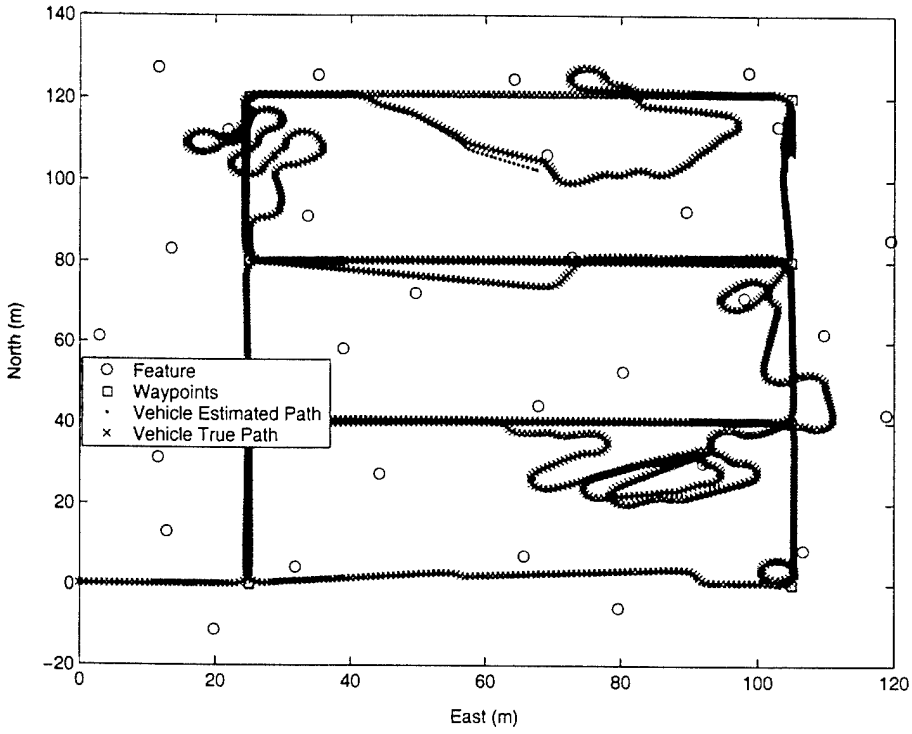


Figure 4-36: Planned Perception, Scenario 3. True vehicle path versus estimated vehicle path. The red crosses represent the true vehicle path. The blue dots follow the estimated vehicle path. The minor jumps in estimated and true position occur when the vehicle obtains an update and improves its estimated position.

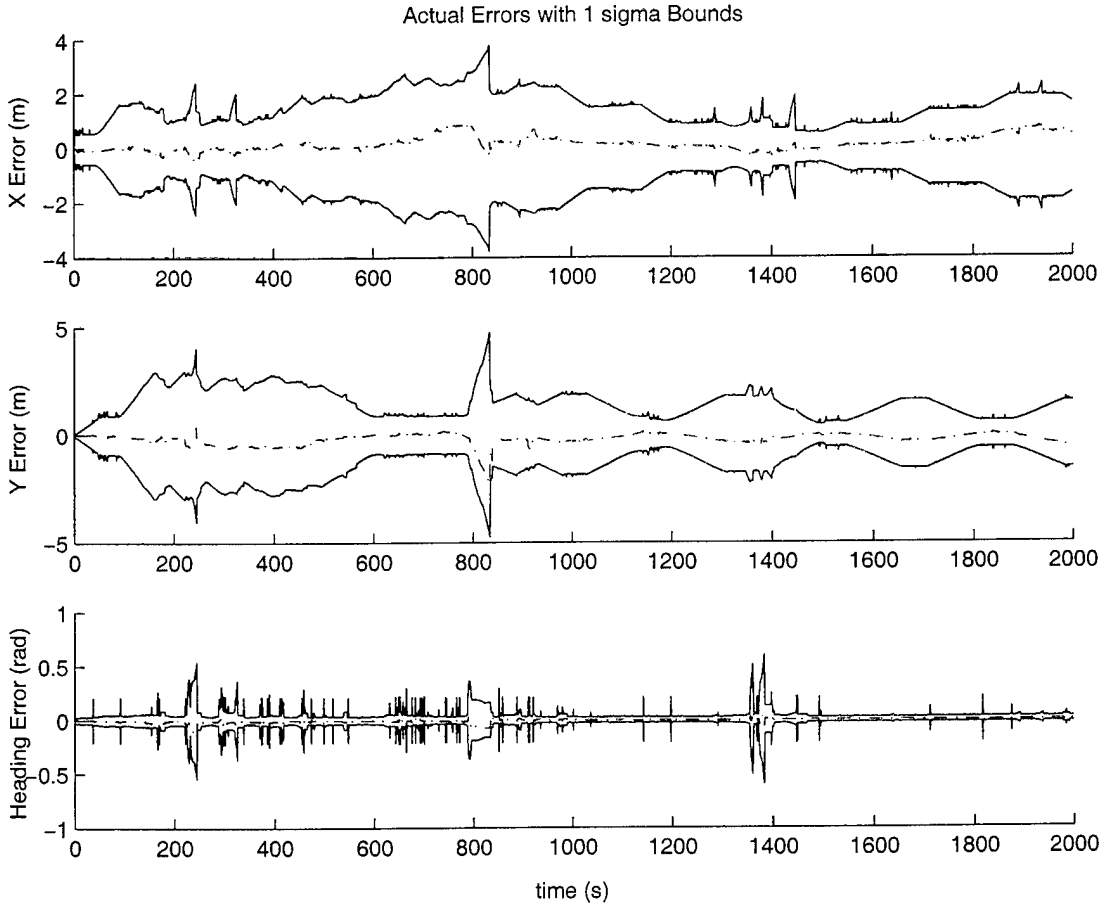


Figure 4-37: Planned Perception, Scenario 3. The errors of the state estimate when performing planned perception. The actual errors are represented by the red dashed line. The solid blue lines represent $\pm 1\sigma$ error bounds. As expected, these errors are bounded.

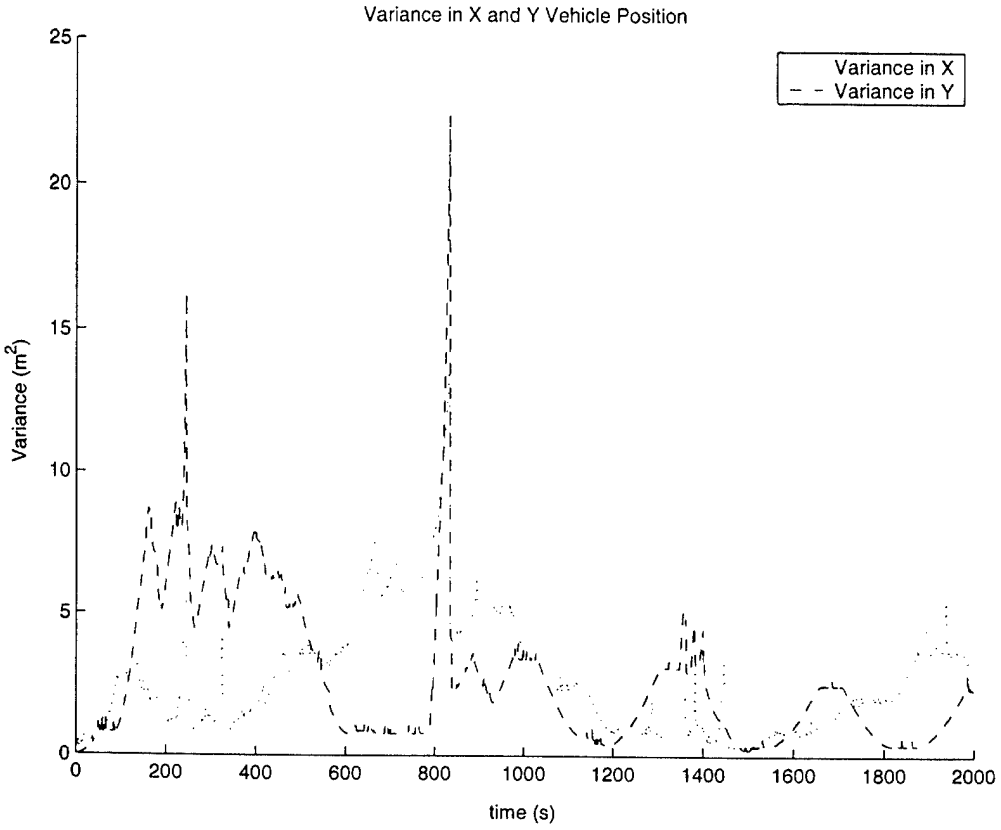
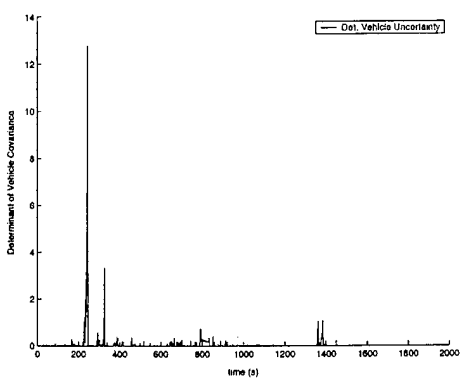
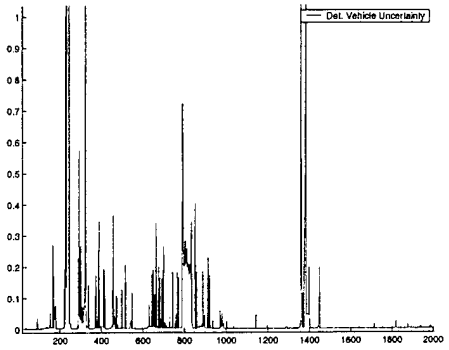


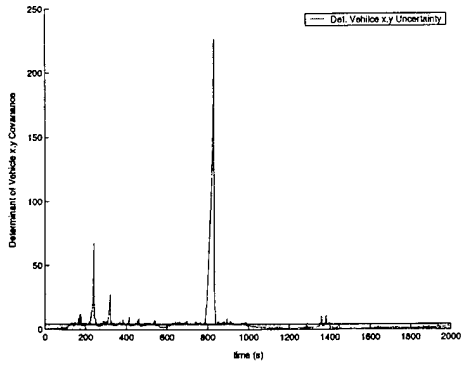
Figure 4-38: Planned Perception, Scenario 3. The variances in x and y position are plotted. The red dotted line is the variance in x while the dashed blue line is the variance in y. As expected, these variances converge to lower limits.



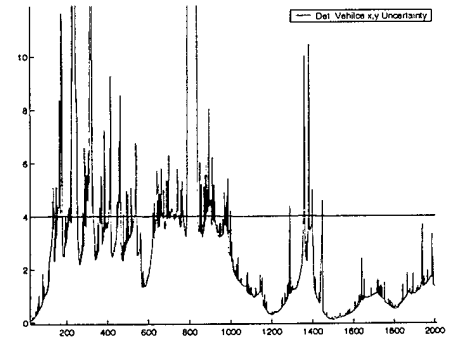
Determinant of Vehicle Uncertainty



Close-up of Determinant

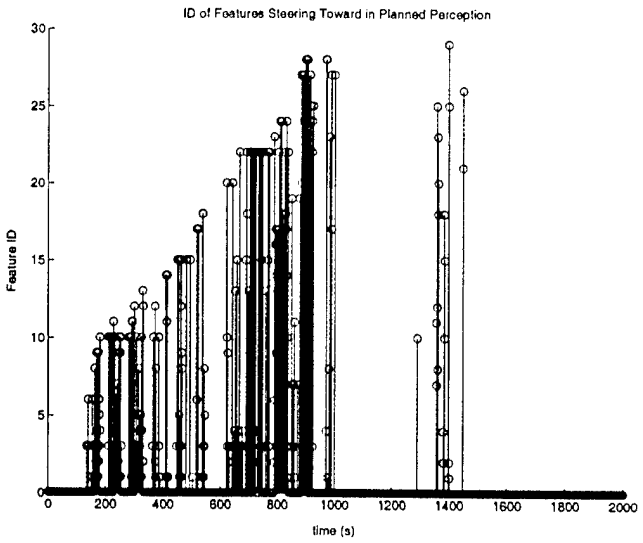


Determinant of Vehicle x,y Uncertainty

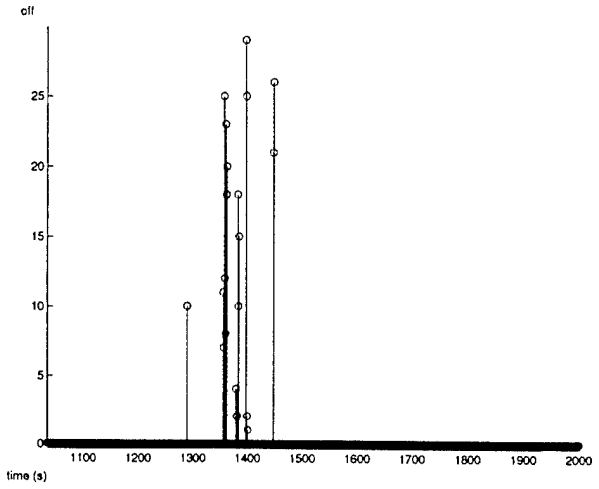


Close-up of Determinant

Figure 4-39: Planned Perception, Scenario 3. **Top left:** Determinant of vehicle uncertainty, $\mathbf{P}_{vv}$. **Top right:** The close-up shows the vehicle uncertainty converging to the initial vehicle uncertainty. **Bottom Left:** The determinant of the vehicle x and y uncertainty, $\mathbf{P}_{xy}$. The black line represents the threshold when planned perception was performed. **Bottom Right:** The uncertainty is quickly restored below the threshold as a result of the planned perception algorithm. The close-up also shows that in the limit, as successive observations are taken, the x and y vehicle uncertainty converges to the initial x,y uncertainty of the vehicle at the time it first observed a feature. This is also below the threshold.



IDs of Features Vehicle Chose to Steer Toward



Close-up of Feature IDs

Figure 4-40: Planned Perception, Scenario 3. **Top:** This figure shows the IDs of the features the vehicle chose to steer toward while performing planned perception. The IDs are obtained as the feature is mapped (i.e. the first feature mapped gets ID #1). The feature ID is 0 when the vehicle is "exploring" and the uncertainty is below the given threshold. **Bottom:** A close-up of the feature ID figure is shown here.

4.3 Conclusions

Navigating by dead reckoning alone requires projecting the vehicle through the vehicle model. There are no observations of features to "reset" the vehicle estimate; no reference can be made to reset the uncertainty in vehicle position. Thus, while navigating by dead reckoning, the absolute position error for dead reckoning grows without bounds as a result of process noise. The unbounded error growth in state estimates are shown in Figure 4-4. Dead reckoning causes the uncertainty associated with the vehicle's covariance matrix to grow exponentially as shown in Figure 4-6.

Concurrent mapping and localization is feature-based navigation. The observation of features is used to obtain localization points for navigation. CML provides a means to navigate so the error estimates remain bounded. This can be seen in Figure 4-8 and Figure 4-9. CML also provides a method for vehicle uncertainty to converge to the initial uncertainty of the vehicle through successive re-observation of features. This is shown in Figure 4-10. As depicted in Figure 4-7, CML provides a more robust way to estimate the vehicle path when compared to dead reckoning. Thus, when navigating from waypoint to waypoint, CML is a substantial improvement to dead reckoning.

Planned perception aims to improve the CML framework. As can be seen in Figure 4-10, the uncertainty in vehicle estimates grows while performing a mission. CML allows for the uncertainty to be reduced through the re-observation of features. However, planned perception allows for this reduction to take place quicker and in a more robust manner. Planned perception constrains vehicle uncertainty. When the uncertainty exceeds a certain threshold, the vehicle maneuvers to re-obtain a reduction in the estimated uncertainty.

The first simulation integrating planned perception and CML focuses on constraining the overall uncertainty of the vehicle, $\mathbf{P}_{vv}$. The first scenario in Section 4.2.3 involves only addressing the first criteria in our planned perception algorithm described in Chapter 3. This scenario focuses on the re-observation of which feature will best improve the uncertainty associated with vehicle state estimates.

This scenario proves to improve CML performance. As shown in Figure 4-10 and

Figure 4-9, while navigating by CML alone, the uncertainties in vehicle estimates grows until the vehicle closes-the-loop or re-observes a feature. When planned perception is integrated in this scenario, CML performance is improved. This can be seen in Figure 4-14 and Figure 4-13. The vehicle is more confident in its estimates than when compared to navigating just by CML. Also, the overall uncertainty in the vehicle is improved. This is seen by comparing Figure 4-10 and Figure 4-14. Thus, integrating planned perception within CML further reduces estimates in vehicle uncertainty.

The second scenario in Section 4.2.3 minimizes the distance between the vehicle and a previously mapped feature. This scenario proves to be a liability to achieving the overall mission objective (moving from waypoint to waypoint). This is because as the uncertainty exceeds the given threshold, the vehicle steers toward the mapped feature closest to it. This may cause "circling" to occur as described in Section 4.2.3. This circling affect is undesirable because re-observing this feature over and over may never allow for the vehicle to reduce the estimated uncertainty below the given threshold. While circling in our experiments, it is by chance that the vehicle happens to re-observe another feature (not the one it is circling) and is able to successfully reduce its uncertainty.

The third scenario combines all three planned perception criteria described in Chapter 3. This presents a more realistic scenario. This is because autonomous vehicles do not have an unlimited power supply. Thus, unlike operating as in the first scenario, it may not always be desirable to travel far distances to re-observe a certain feature. Therefore, by incorporating all criteria, a trade-off between each is presented. The feature which best satisfies all three criteria is the one chosen in the planned perception algorithm. Like the first scenario, this simulation proves to be an improvement to CML. The uncertainty associated in the vehicle estimates is improved as shown by comparing Figure 4-10 and Figure 4-24.

The integration of planned perception and CML described in the above scenarios provide methods to alter the vehicle's motion to maintain an uncertainty level. However, by constraining the determinant of $\mathbf{P}_{vv}$, the uncertainty in the x and y position

is still relatively large. This can be seen in determinant figures and variance plots of the above scenarios. Performing planned perception by constraining $\mathbf{P}_{vv}$ proves to improve CML. However, the question is then raised if it is possible to obtain an improvement to the vehicle's x and y uncertainty.

The second simulation addresses the question of constraining the uncertainty in the vehicle's x and y estimates; it constrains the associated determinant of $\mathbf{P}_{xy}$. The first scenario in Section 4.2.4 proves to drastically improve CML performance. This is shown in Figure 4-29. By constraining the uncertainty in the x and y vehicle estimates, the uncertainty in the overall vehicle estimate, $\mathbf{P}_{vv}$, is also reduced. This can be seen by comparing Figure 4-10 and Figure 4-29.

The second scenario in Section 4.2.4 again proves that performing planned perception by relying on the re-observation of the nearest feature may not achieve the mission objective. Again, "circling" becomes an issue. This is seen by the loops in the vehicle path of Figure 4-31 and by analyzing the features the vehicle steers toward in Figure 4-35.

By constraining vehicle uncertainty, planned perception improves the CML framework. It provides a method to adaptively alter the vehicle's sensing strategy in order to maintain a better estimate of vehicle states. Two methods of constraining vehicle uncertainty are introduced. The first involves constraining the overall uncertainty associated with the vehicle, $\mathbf{P}_{vv}$. The second constrains the uncertainty associated with the x and y estimates of vehicle position, $\mathbf{P}_{xy}$. Constraining $\mathbf{P}_{xy}$ proves to be more desirable.

There is, however, a trade-off between the two constraints. By constraining $\mathbf{P}_{xy}$, the vehicle spends most of its "first lap" of the pattern operating in *localization mode* performing planned perception. Once the vehicle closes-the-loop, it is able to maintain a smaller uncertainty level during the rest of its mission. Thus, more energy is spent localizing the vehicle in the beginning of the mission when constraining $\mathbf{P}_{xy}$. After closing-the-loop, the vehicle operates with more certainty compared to performing planned perception constraining $\mathbf{P}_{vv}$.

A trade-off also exists between the weighting gains and the three different planned

perception criteria. Addressing the first criteria alone yields the best planned perception performance. However, it may not always be desirable, due to energy costs, to re-visit features that are farther away than others. Also, only addressing the second criteria proves to cause the vehicle to "circle" features. This circling and re-observation of the same feature over and over does not always yield the needed update to reduce vehicle uncertainty. Thus, by incorporating all three criteria, the vehicle no longer continually "circles" certain features and is not required to travel back to the first features mapped. The algorithm allows for the weighting gains to be set, determining the performance of the planned perception algorithm, according to the specific, desired mission.

4.4 Summary

This chapter presented the description of the simulation written in © MATLAB combining the ideas presented in Chapter 2 and Chapter 3. The results integrating planned perception within CML were also presented. The results and performances of the different simulations were then compared.

Chapter 5

Conclusions and Future Research

This chapter summarizes the contributions of this thesis and presents suggestions for future research.

5.1 Contributions

This thesis presents a method for integrating planned perception within concurrent mapping and localization. Planned perception is the process of adaptively determining the sensing strategy of the mobile robot. The goal of integrating planned perception with CML is to provide the mobile robot with a means to determine the optimal action given the current knowledge of robot pose, sensors, and the environment. CML performance, augmented with smarter sensing strategies, exhibits improved performance by motivating changes in robot orientation and sensing strategies. Our planned perception algorithm aims to minimize the uncertainty associated with the vehicle's state estimates. It provides a method that addresses the trade-off between mission success and knowledge of the vehicle's current state. Our approach was applied and validated in simulation.

5.2 Future Research

This thesis is motivated by the study of autonomous underwater vehicles (AUVs). Thus, a suggested area for future research includes integrating planned perception within CML in the real world with actual autonomous robots. Extending the ideas presented in this thesis to real world environments might require insight and future research in the areas of vehicle modelling, data association, and navigating by CML in dynamic environments.

One extension to planned perception may be to incorporate sensor characteristics into adaptive sensing strategies. Taking into account the sensor's range and field of view may motivate the vehicle steer not towards certain features, but towards a "virtual waypoint" that would allow the vehicle to observe certain features. Another extension may be to develop a policy that explicitly minimizes angular error in the map. This may be performed by analyzing the estimated angular error between features and developing a strategy whose motion reduces this error.

Our approach provides a method to minimize the uncertainty associated in the vehicle estimates. An extension of integrating planned perception and CML may be to analyze a different metric of performing adaptive sensing; instead of utilizing a policy to re-observe features, utilize a policy that integrates adaptive motion control and CML based on the correlation structure of the covariance matrix. Therefore, instead of choosing to re-observe a feature that reduces vehicle uncertainty, choose actions that make the map more fully correlated.

Bibliography

[1] W. Au. *The Sonar of Dolphins.* New York: Springer-Verlag, 1993.

[2] N. Ayache and O. Faugeras. Maintaining representations of the environment of a mobile robot. *IEEE Trans. Robotics and Automation*, 5(6):804–819, 1989.

[3] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association.* Academic Press, 1988.

[4] Y. Bar-Shalom and X. R. Li. *Estimation and Tracking: Principles, Techniques, and Software.* Yaakov Bar-Shalom (YBS), 1998.

[5] P. R. Belanger. *Control Engineering A Modern Approach.* Saunders College Publishing, 1995.

[6] J. G. Bellingham and J. S. Willcox. Optimizing AUV oceanographic surveys. In *IEEE Conference on Autonomous Underwater Vehicles*, Monterey, CA, 1996.

[7] R. A Brooks. Aspects of mobile robot visual map making. In *Second Int. Symp. Robotics Research*, Tokyo, Japan, 1984. MIT Press.

[8] R. Brown and P. Hwang. *Introduction to Random Signals and Applied Kalman Filtering.* J. Wiley and Sons, 1992.

[9] W. Burgard, A.B. Cremers, D. Fox, D. Haehnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 1999.

[10] R. N. Carpenter and M. R. Medeiros. Concurrent mapping and localization and map matching on autonomous underwater vehicles. *Oceans*, 2001.

[11] J. A. Castellanos, J. D. Tardos, and G. Schmidt. Building a global map of the environment of a mobile robot: The importance of correlations. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1053–1059, 1997.

[12] CNN ©. *Death of the Kursk*. http://www.cnn.com/SPECIALS/2000/submarine/, August 21, 2000.

[13] I. J. Cox. Blanche-an experiment in guidance and naviagion of an autonomous robot vehicle. pages 423–428, 1996.

[14] A. J. Davison and D. W. Murray. Mobile robot laclisation using active vision. ECCV, 1998.

[15] A. J. Davison. *Mobile Robot Navigation Using Active Vision*. Ph.D. thesis, University of Oxford, 1999.

[16] X. Deng and A. Mirzaian. Competitive robot mapping with homogeneous markers. *IEEE Transactions on Robotics and Automation*, 12:532-542, Aug. 1996.

[17] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, June 2001.

[18] H. J. S. Feder, J. J. Leonard, and C. M. Smith. Adaptive concurrent mapping and localization using sonar. In *Proc. IEEE Int. Workshop on Intelligent Robots and Systems*, Victoria, B.C., Canada, October 1998.

[19] H. J. S. Feder, J. J. Leonard, and C. M. Smith. Adaptive sensing for terrain aided navigation. In *IEEE Oceans*, Nice, France, September 1998.

[20] H. J. S. Feder, J. J. Leonard, and C. M. Smith. Adaptive mobile robot navigation and mapping. *Int. J. Robotics Research*, 18(7):650–668, July 1999.

[21] H. J. S. Feder. *Simultaneous Stochastic Mapping and Localization.* Ph.D. thesis, MIT Department of Ocean Engineering, 1999.

[22] J. W. Fenwick. *Collaborative Concurrent Mapping and Localization.* M.S. thesis, MIT Department of Electrical Engineering and Computer Science, May 2001.

[23] A. C. Gelb. *Applied Optimal Estimation.* The MIT Press, 1973.

[24] J. E. Giuvant and E. M. Nebot. Optimization of the simultaneous localizatin and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, June 2001.

[25] B. J. Kuipers and Y. Byun. A robot exploration and mapping strategy based on a semantic heirarchy of spatial representations. *Robotics and Autonomous Systems*, pages 47–63, 1991.

[26] J. J. Leonard, A. A. Bennett, C. M. Smith, and H. J. S. Feder. Autonomous underwater vehicle navigation. *IEEE ICRA Workshop on Navigation of Outdoor Autonomous Vehicles*, 1998.

[27] J. J. Leonard, R. N. Carpenter, and H. J. S. Feder. Stochastic mapping using forward look sonar. In *International Conference on Field and Service Robotics*, pages 69–74, Pittsburgh, Pennsylvania, August 1999.

[28] J. J. Leonard and H. F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Proc. IEEE Int. Workshop on Intelligent Robots and Systems*, pages 1442–1447, Osaka, Japan, 1991.

[29] J. J. Leonard and H. F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation.* Boston: Kluwer Academic Publishers, 1992.

[30] J. J. Leonard and H. J. S. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In D Koditschek and J. Hollerbach, editors, *Robotics Research: The Ninth International Symposium*, pages 169–176, Snowbird, Utah, 2000. Springer Verlag.

[31] J. J. Leonard. *Directed Sonar Sensing for Mobile Robot Navigation.* PhD thesis, University of Oxford, 1990.

[32] J. S. Manyika and H. F. Durrant-Whyte. *Data Fusion and Sensor Management: A decentralized information-theoretic approach.* New York: Ellis Horwood, 1994.

[33] P. Maybeck. *Stochastic Models, Estimation, and Control, vol. 1.* Academic Press, 1979.

[34] P. H. Milne. *Underwater Acoustic Positioning Systems.* Spon Ltd, New York, 1983.

[35] H. Moravec. Sensor fusion in certainty grids for mobile robots. In *Sensor Devices and Systems for Robotics*, pages 253–276. Springer-Verlag, 1989. Nato ASI Series.

[36] U. S. Navy. *The Navy Unmanned Undersea Vehicle (UUV) Master Plan.* Approved for Public Release 27 March 2001, April 20, 2000. http://www.auvsi.org/resources/UUVMPPubRelease.pdf.

[37] J. Neira and J. D. Tardos. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6), Dec 2001.

[38] P. M. Newman. *On the structure and solution of the simultaneous localization and mapping problem.* PhD thesis, University of Sydney, 1999.

[39] N. Polmar. *Naval Institute Guide to the Ships and Aircraft of the U. S. Fleet, 17th Ed.* Naval Institute Press, Annapolis, MD, 2001.

[40] U.S. Government Resources. *US Navy's Submarine Rescue Team.* http://usgovinfo.about.com/library/weekly/aa081700a.htm, Dateline: August 17, 2000.

[41] R. J. Rikoski, J. J. Leonard, and P. M. Newman. Stochastic mapping frameworks. 2001.

[42] H. Singh. *An entropic framework for AUV sensor modelling.* PhD thesis, Massachusetts Institute of Technology, 1995.

[43] G. M. Siouris. *An Engineering Approach to Optimal Control and Estimation Theory.* John Wiley & Sons, Inc., New York, 1996.

[44] C. Smith. *Integrating mapping and navigation.* Ph.D. thesis, MIT Department of Ocean Engineering, 1998.

[45] R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In *4th International Symposium on Robotics Research.* MIT Press, 1987.

[46] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*, pages 167–193. Springer-Verlag, 1990.

[47] A. K. Sood and H. Wechsler. *Active Perception and Robot Vision.* Germany: Springer-Verlag Berlin Heidelberg, 1992.

[48] S. Thrun, A. Bucken, W. Burgard, D. Fox, T. Frohlinghaus, D. Hennig, T. Hofmann, M. Krell, and T. Schmidt. Map learning and high-speed navigation in rhino. MIT/AAAI Press, 1998.

[49] S. Thrun, J.-S. Gutmann, D. Fox, W. Bugard, and B. J. Kuipers. Integrating topological and metric maps for mobile robot navigation: A statistical approach. In *AAAI-98*, 1998.

[50] S. Thrun. A probabilistic on-line mapping algorithm for teams of mobile robots. *The International Journal of Robotics Research*, 20(5):335–363, May 2001.

[51] J. Vaganay, J. G. Bellingham, and J. J. Leonard. Outlier rejection for autonomous acoustic navigation. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 2174–2181, April 1996.

[52] L. Whitcomb, D. R. Yoerger, H. Singh, and J. Howland. Advances in underwater robot vehicles for deep ocean exploration: Navigation, control, and survery operations.

[53] S. B. Williams. *Efficient Solutions to Autonomous Mapping and Navigation Problems.* Ph.D. thesis, The University of Sydney, 2001.